Atmel mXT640U Revision 1.0

maXTouch 640-node Touchscreen Controller

DATASHEET

Features

- Atmel® maXTouch® Adaptive Sensing Touchscreen Technology
  - Up to 32 X (transmit) lines and 20 Y (receive) lines
  - A maximum of 640 nodes can be allocated to the touchscreen
  - Screen sizes of 6.2 inches diagonal are supported (subject to configuration)
  - Larger screen sizes for Android applications supported (subject to configuration)
  - Multi-touch support with up to 16 concurrent touches tracked in real time
- Dual-boot OS support for Microsoft® Windows® and Android
- Advanced Touch Handling
  - Moisture/Water Compensation
    - No false touch with condensation or water drop up to 22 mm diameter
    - One-finger tracking with condensation or water drop up to 22 mm diameter
  - Stylus Support
    - Supports passive stylus with 1 mm contact diameter, subject to configuration, stack up, and sensor design
  - Glove Support
    - Supports multiple-finger glove touch up to 1.5 mm thickness
    - Supports single-touch gloved operation with various materials up to 5 mm thickness
- Touch Performance
  - Mutual capacitance and self capacitance measurements supported for touch detection
  - Response Times
    - Initial latency <20 ms for first touch from idle, subject to configuration
  - Atmel maXCharger® technology to combat ambient, charger noise, and power-line noise
    - Up to 240 Vpp between 1 Hz and 1 kHz sinusoidal waveform
    - Up to 20 Vpp between 1 kHz and 1 MHz sinusoidal waveform
  - Scan Speed
    - Typical report rate for 10 touches ≥100 Hz (subject to configuration)
    - Configurable to allow for power and speed optimization
    - Programmable timeout for automatic transition from active to idle states
- Enhanced Algorithms
  - Lens bending algorithms to remove signal distortions
  - Touch suppression algorithms to remove unintentional touches, such as face or palm
  - Palm Recovery Algorithm for quick restoration to normal state
  - Supports wake up/unlock gestures, including symbol recognition

- Panel / Cover Glass Support
  - Supports fully-laminated sensors, touch-on-lens stack-ups and on-cell designs
  - Works with PET or glass, including curved profiles
    - Glass from 0.5 to 2.5 mm, dependent on screen size, touch size, and stack-up
    - Plastic from 0.25 mm to 1.2 mm, dependent on screen size and touch size
  - Works with all proprietary sensor patterns recommended by Atmel
- Keys
  - Up to 32 nodes can be allocated as mutual capacitance sensor keys (subject to other configurations)
  - Support for 3 Generic Keys in addition to the touchscreen array (subject to other configurations)
  - Adjacent Key Suppression® (AKS®) technology is supported for false key touch prevention
- Product Data Store Area
  - Up to 32 bytes of user-defined data can be stored during production
- Power Saving
  - Programmable timeout for automatic transition from active to idle states
  - Pipelined analog sensing detection and digital processing to optimize system power efficiency
- Application Interfaces
  - I^2C-compatible slave mode: Standard/Fast mode 400 kHz, Fast-plus mode 1 MHz, High-speed mode up to 3.4 MHz
  - HID-I^2C interface for Microsoft® Windows® 8.x
  - Interrupt to indicate when a message is available
- Power Supply
  - Digital (Vdd) 3.3 V nominal
  - Analog (AVdd) 3.3 V nominal
  - Host interface I/O voltage (VddIO) 1.8 V to 3.3 V nominal
  - High voltage internal X line drive (XVdd) 9.9 V, with internal voltage pump
- Package
  - 88-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm pitch
  - 88-ball X1FBGA 6 × 6 × 0.45 mm, 0.5 mm pitch
- Environmental Conditions
  - Operating temperature –40°C to +85°C

Table of Contents

1. Overview of mXT640U

The Atmel maXTouch family of touch controllers brings industry-leading capacitive touch performance to customer applications. The mXT640U features the latest generation of Atmel Adaptive Sensing technology that utilizes a hybrid mutual- and self-capacitive sensing system in order to deliver unparalleled touch features and a robust user experience.

- **Patented capacitive sensing method** – The mXT640U uses a unique charge-transfer acquisition engine to implement the Atmel-patented QMatrix® capacitive sensing method. Coupled with a state-of-the-art CPU, the entire touchscreen sensing solution can measure, classify and track a number of individual finger touches with a high degree of accuracy in the shortest response time.
- **Capacitive Touch Engine (CTE)** – The mXT640U features an acquisition engine, which uses an optimal measurement approach to ensure almost complete immunity from parasitic capacitance on the receiver input lines. The engine includes sufficient dynamic range to cope with anticipated touchscreen self and mutual capacitances, which allows great flexibility for use with the Atmel proprietary sensor pattern designs. One- and two-layer ITO sensors are possible using glass or PET substrates.
- **Touch detection** – The mXT640U allows for both mutual- and self-capacitance measurements, with the self-capacitance measurements being used to augment the mutual-capacitance measurements to produce reliable touch information.

  When self-capacitance measurements are enabled, touch classification is achieved using both mutual- and self-capacitance touch data. This has the advantage that both types of measurement systems can work together to detect touches under a wide variety of circumstances.

  During idle mode, the device performs self-capacitance touch scans. When a touch is detected, the device starts performing mutual capacitance touch scans as well as self capacitance scans.

  Mutual-capacitance touch data is used wherever possible to classify touches as this has greater granularity than self-capacitance measurements and provides positional information on touches. For this reason, multiple touches can only be determined by mutual-capacitance touch data. If the self-capacitance touch processing detects multiple touches, touchscreen processing is skipped until mutual-capacitance touch data is available.

  Self-capacitance measurements, on the other hand, allow for the detection of single touches in extreme cases, such as single thick-glove touches, when touches can only be detected by self-capacitance data and may be missed by mutual-capacitance touch detection.
- **Display Noise Cancellation** – A combination of analog circuitry, hardware noise processing, and firmware that combats display noise without requiring additional listening channels or synchronization to display timing. This enables the use of shieldless touch sensor stacks, including touch-on-lens.
- **Noise filtering** – Hardware noise processing in the capacitive touch engine provides enhanced autonomous filtering and allows a broad range of noise profiles to be handled. The result is good performance in the presence of charger and LCD noise.
- **Processing power** – The main CPU has two powerful microsequencer coprocessors under its control consuming low power. This system allows the signal acquisition, preprocessing, postprocessing and housekeeping to be partitioned in an efficient and flexible way.
- **Interpreting user intention** – The Atmel hybrid mutual- and self-capacitance method provides unambiguous multitouch performance. Algorithms in the mXT640U provide optimized touchscreen position filtering for the smooth tracking of touches, responding to a user's intended touches while preventing false touch triggered by ambient noise or conductive material on the sensor surface, such as water. The suppression of unintentional touches from the user’s gripping fingers, resting palm or touching cheek or ear also help ensure that the user’s intentions are correctly interpreted.

2. Connection and Configuration Information

2.1 Pin Configuration – UFBGA/X1FBGA 88 Balls

	1	2	3	4	5	6	7	8	9	10	11
A	AVDD	DS0	Y18	Y16	Y14		Y8	Y6	Y4	Y2	Y0
B	X18		Y19	Y17	Y15		Y7	Y5	Y3		AVDD
C	X20	X19		GND	Y13		Y9	Y1		X0	X1
D	X22	X21	X17		Y12	Y11	Y10		GND	X2	X3
E	X24	X23	X25	X26				X7	X6	X4	X5
F				X27				X8			
G	X30	X31	X29	X28				X9	X10	X12	X11
H	RESV	RESV	EXTCAP1		GPIO2 DBG_CLK	$\overline{\text{TEST}}$	$\overline{\text{CHG}}$		GND	X14	X13
J	EXTCAP0	EXTCAP3		GND	$\overline{\text{GPIO1}}$ $\overline{\text{DBG_SS}}$ DBG2_FRAME		ADDSEL DBG2_DATA0	GPIO6 DBG_DAT DBG2_DATA5		X16	X15
K	EXTCAP2		VDDIO	$\overline{\text{RESET}}$	GPIO0 DBG2_CLK		I2CMODE DBG2_DATA2	GPIO5 DBG2_DATA4	GKEYY2		GKEYX0
L	XVDD	VDD	VDDCORE	SCL	SDA		SYNC GPIO3 DBG2_DATA1	NOISE_IN GPIO4 DBG2_DATA3	GKEYY1	GKEYY0	XVDD

Top View

Table 2-1. Pin Listing UFBGA/X1FBGA 88 Balls

Ball	Name	Type	Comments	If Unused, Connect To...
A1	AVDD	P	Analog power	–
A2	DS0	S	Driven Shield signal; used as guard track between X/Y signals and ground	Leave open
A3	Y18	S	Y line connection	Leave open
A4	Y16	S	Y line connection	Leave open
A5	Y14	S	Y line connection	Leave open
–				
A7	Y8	S	Y line connection	Leave open
A8	Y6	S	Y line connection	Leave open
A9	Y4	S	Y line connection	Leave open
A10	Y2	S	Y line connection	Leave open
A11	Y0	S	Y line connection	Leave open
B1	X18	S	X line connection	Leave open
–				
B3	Y19	S	Y line connection	Leave open
B4	Y17	S	Y line connection	Leave open
B5	Y15	S	Y line connection	Leave open
–				
B7	Y7	S	Y line connection	Leave open
B8	Y5	S	Y line connection	Leave open
B9	Y3	S	Y line connection	Leave open
–				
B11	AVDD	P	Analog power	–
C1	X20	S	X line connection	Leave open
C2	X19	S	X line connection	Leave open
–				
C4	GND	P	Ground	–
C5	Y13	S	Y line connection	Leave open
–				
C7	Y9	S	Y line connection	Leave open
C8	Y1	S	Y line connection	Leave open
–				
C10	X0	S	X line connection	Leave open

Table 2-1. Pin Listing UFBGA/X1FBGA 88 Balls (Continued)

Ball	Name	Type	Comments	If Unused, Connect To...
C11	X1	S	X line connection	Leave open
D1	X22	S	X line connection	Leave open
D2	X21	S	X line connection	Leave open
D3	X17	S	X line connection	Leave open
–				
D5	Y12	S	Y line connection	Leave open
D6	Y11	S	Y line connection	Leave open
–				
D7	Y10	S	Y line connection	Leave open
–				
D9	GND	P	Ground	–
D10	X2	S	X line connection	Leave open
D11	X3	S	X line connection	Leave open
E1	X24	S	X line connection	Leave open
E2	X23	S	X line connection	Leave open
E3	X25	S	X line connection	Leave open
E4	X26	S	X line connection	Leave open
–				
E8	X7	S	X line connection	Leave open
E9	X6	S	X line connection	Leave open
E10	X4	S	X line connection	Leave open
E11	X5	S	X line connection	Leave open
–				
F4	X27	S	X line connection	Leave open
F8	X8	S	X line connection	Leave open
G1	X30	S	X line connection	Leave open
G2	X31	S	X line connection	Leave open
G3	X29	S	X line connection	Leave open
G4	X28	S	X line connection	Leave open
–				
G8	X9	S	X line connection	Leave open
G9	X10	S	X line connection	Leave open
G10	X12	S	X line connection	Leave open

Table 2-1. Pin Listing UFBGA/X1FBGA 88 Balls (Continued)

Ball	Name	Type	Comments	If Unused, Connect To...
G11	X11	S	X line connection	Leave open
H1	RESV	S	Reserved for future use	Leave open
H2	RESV	S	Reserved for future use	Leave open
H3	EXTCAP1	P	Connect to EXTCAP2 via capacitor; see Section 3.2 on page 14	Leave open
–			–	
H5	GPIO2 DBG_CLK	I/O O	General purpose I/O Primary Debug clock	Input: GND Output: leave open
H6	$\overline{\text{TEST}}$	–	Reserved for factory use; pull up to VDDIO	Pull up to VDDIO
–			–	
H7	$\overline{\text{CHG}}$	OD	State change interrupt. Pull up to VddIO	Pull up to VddIO
–			–	
H9	GND	P	Ground	–
H10	X14	S	X line connection	Leave open
H11	X13	S	X line connection	Leave open
J1	EXTCAP0	P	Connect to EXTCAP3 via capacitor; see Section 3.2 on page 14	Leave open
J2	EXTCAP3	P	Connect to EXTCAP0 via capacitor; see Section 3.2 on page 14	Leave open
–			–	
J4	GND	P	Ground	–
J5	GPIO1 $\overline{\text{DBG\_SS}}$ DBG2_FRAME	I/O O O	General Purpose I/O Primary Debug SS line; pull up to VddIO Secondary Debug Frame; for more information see Section 13. on page 55	Input: GND Output: leave open
–			–	
J7	ADDSEL DBG2_DATA0	I O	I2C address select; see Section 8.2 on page 29 Secondary Debug Data 0; for more information see Section 13. on page 55	–
J8	GPIO6 DBG_DAT DBG2_DATA5	I/O O O	General purpose I/O Primary Debug data Secondary Debug Data 5; for more information see Section 13. on page 55	Input: GND Output: leave open
–			–	
J10	X16	S	X line connection	Leave open
J11	X15	S	X line connection	Leave open

Table 2-1. Pin Listing UFBGA/X1FBGA 88 Balls (Continued)

Ball	Name	Type	Comments	If Unused, Connect To...
K1	EXTCAP2	P	Connect to EXTCAP1 via capacitor; see Section 3.2 on page 14	Leave open
–				
K3	VDDIO	P	Digital IO interface power	–
K4	$\overline{\text{RESET}}$	I	Connection to host system is recommended	Pull up to VDDIO
K5	GPIO0 DBG2_CLK	I/O O	General purpose I/O Secondary Debug Clock; for more information see Section 13. on page 55	Input: GND Output: leave open
–				
K7	I2CMODE DBG2_DATA2	I O	Selects I2C mode; see Section 8. on page 28 Secondary Debug Data 2; for more information see Section 13. on page 55	–
K8	GPIO5 DBG2_DATA4	I/O O	General purpose I/O Secondary Debug Data 4; for more information see Section 13. on page 55	Input: GND Output: leave open
K9	GKEYY2	S	Y line connection	Leave open
–				
K11	GKEYX0	S	X line connection	Leave open
L1	XVDD	P	X line drive power	–
L2	VDD	P	Digital Power	–
L3	VDDCORE	P	Digital core power	–
L4	SCL	OD	Serial Interface Clock	–
L5	SDA	OD	Serial Interface Data	–
–				
L7	SYNC GPIO3 DBG2_DATA1	I I/O O	Measurement synchronization input General purpose I/O Secondary Debug Data 1; for more information see Section 13. on page 55	Input: GND Output: leave open
L8	NOISE_IN GPIO4 DBG2_DATA3	I I/O O	Noise present input General purpose I/O Secondary Debug Data 3; for more information see Section 13. on page 55	Input: GND Output: leave open
L9	GKEYY1	S	Y line connection	Leave open

Table 2-1. Pin Listing UFBGA/X1FBGA 88 Balls (Continued)

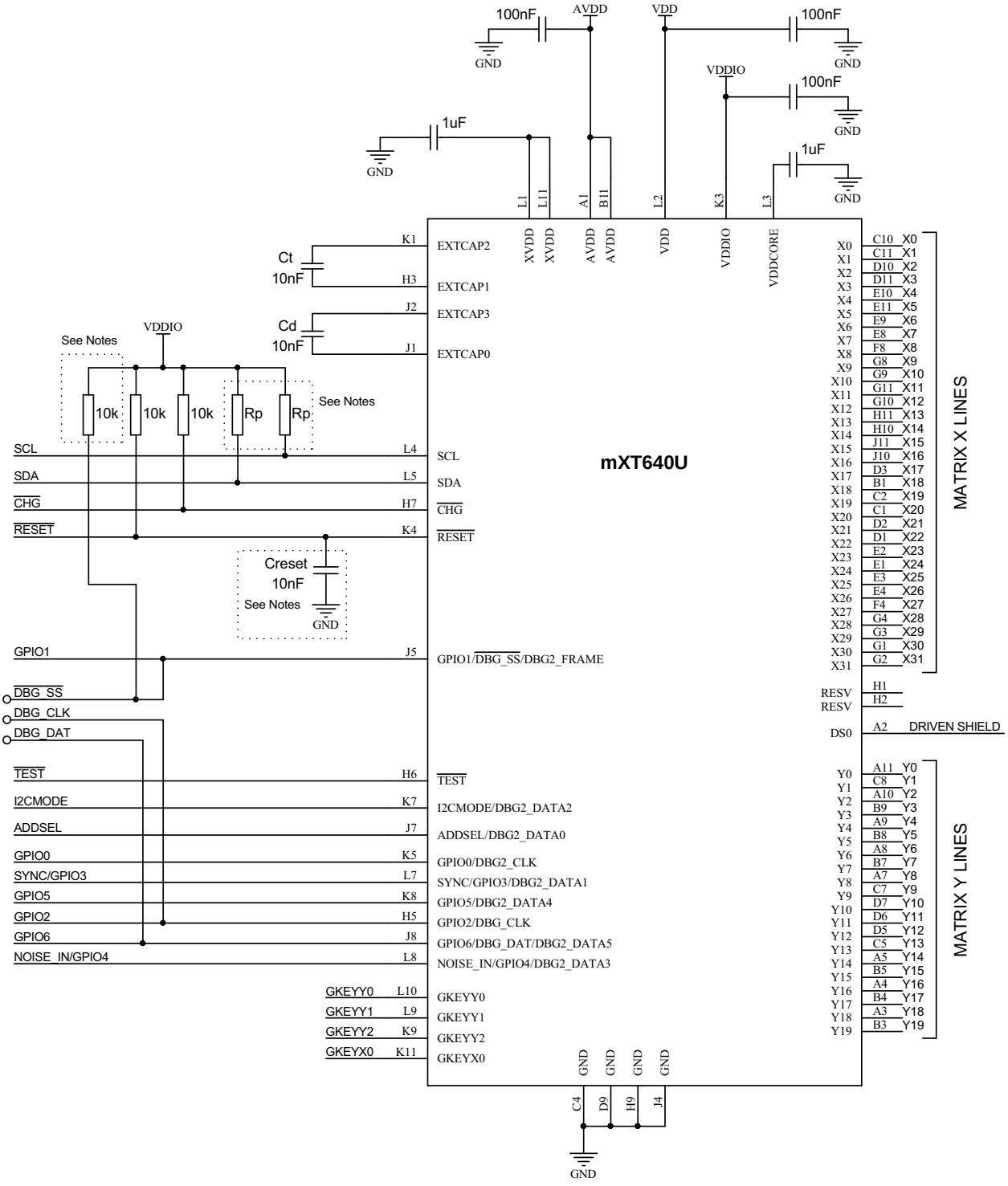
Ball	Name	Type	Comments	If Unused, Connect To...
L10	GKEYY0	S	Y line connection	Leave open
L11	XVDD	P	X line drive power	–

Key:

I	Input only	O	Output only	I/O	Input or output
OD	Open drain output	P	Ground or power	S	Sense pin

3. Schematic

3.1 Schematic UFBGA/X1FBGA 88 Balls



See Section 3.2 “Schematic Notes” on page 14

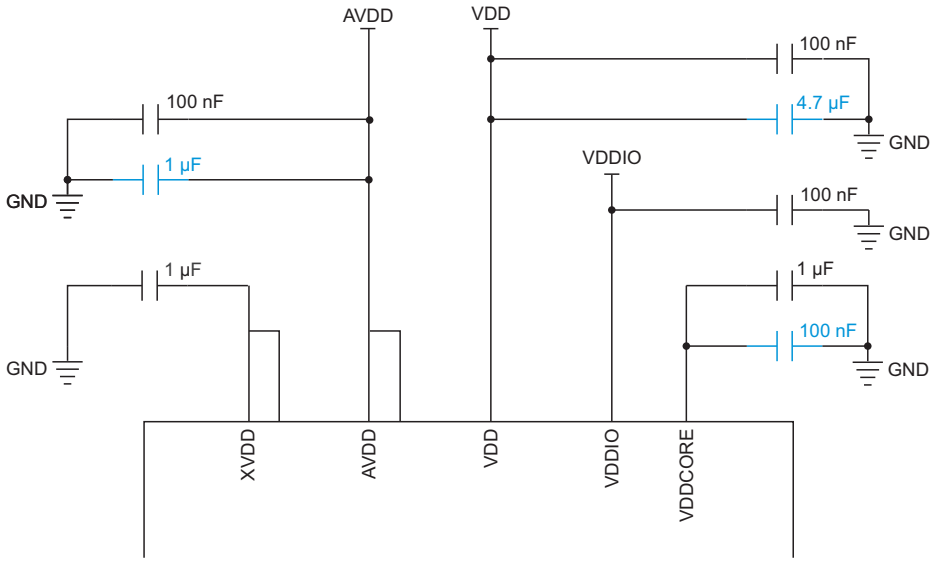
3.2 Schematic Notes

CAUTION: The device may be permanently damaged if any XVDD pin is shorted to ground or high current is drawn from it.

3.2.1 Decoupling capacitors

1. Decoupling capacitors must be X7R or X5R and placed <5 mm away from the pins for which they act as bypass capacitors.
2. The schematics show the minimum capacitors required. If the ball configuration means that sharing a bypass capacitor is not possible, then the number of capacitors should be increased.
3. If the device is placed on the system board, the minimum number of capacitors required is as shown in the schematic on page 13. Note that this requires that the voltage regulator supplies for AVDD/VDD and VDDIO are clean and noise free. It also assumes that the track length between the capacitors and on-board power supplies is < 50 mm.
4. If an active tail design is used, the voltage regulators are likely to be some distance from the device and it may be necessary to implement additional decoupling. In this case, a parallel combination of capacitors is recommended to give high and low frequency filtering, as shown in Figure 3-1.

Figure 3-1. Additional Recommended Decoupling Capacitors



NOTE: Recommended additional decoupling capacitors are shown in blue

3.2.2 Internal Voltage Pump

The voltage pump requires two external capacitors as follows:

- EXTCAP0 must be connected to EXTCAP3 via a capacitor (Cd)
- EXTCAP1 must be connected to EXTCAP2 via a capacitor (Ct)

Capacitors Cd and Ct should each provide a capacitance of 10 nF. The value may be varied under certain circumstances after consultation with Atmel.

3.2.3 VDDCORE

VddCore is internally generated from the Vdd power supply. To guarantee stability of the internal voltage regulator, an external capacitor is required. The capacitor should have a value of 1 µF.

3.2.4 $\overline{\text{RESET}}$ Line

The $\overline{\text{RESET}}$ line is shown on the schematics with a 10 nF capacitor (Creset) to ground. This capacitor is optional but may help if ESD issues are encountered.

3.2.5 I^2C Interface

The resistors labeled Rp are pull-up resistors for the SDA and SCL lines. The values of these resistors depends on the speed of the I^2C interface. See Section 14.9 on page 64 for details. Note that if a VddIO supply at the low end of the allowable range is used, the pull-up resistor values may need to be reduced.

3.2.6 Multiple Function Pins

Some pins may have multiple functions. In this case, only one function can be chosen and the circuit should be designed accordingly.

3.2.7 GPIO Pins

The mXT640U has 7 GPIO pins. The pins can be set to be either an input or an output, as required, using the GPIO/PWM Configuration T19 object (refer to the *mXT640U 1.0 Protocol Guide*).

Unused GPIO pins can be left externally unconnected as long as they are given a defined state by using the GPIO/PWM Configuration T19 object. By default GPIO pins are set to be inputs and if they are not used they should be connected to GND. Alternatively, they can be set as outputs using the GPIO/PWM Configuration T19 object and left open.

If the GPIO/PWM Configuration T19 object is not enabled for use, all the GPIO pins are unused.

Some GPIO pins have alternative functions. If an alternative function is used then this takes precedence over the GPIO function and the pin cannot be used as a GPIO pin. In particular:

- GPIO4 cannot be used if the NOISE_IN function is in use
- GPIO3 cannot be used if the SYNC function is in use
- The SPI Debug Interface functionality is shared with some of the GPIO pins. If the SPI Debug Interface is in use, these pins cannot be used as GPIO pins (see Section 3.2.8). Note that if these GPIO pins are totally unused, they should be set as outputs (refer to the GPIO/PWM Configuration T19 object in the *mXT640U 1.0 Protocol Guide*).

3.2.8 DBG_CLK, DBG_DATA and $\overline{\text{DBG\_SS}}$ Lines

The DBG_CLK, DBG_DATA and $\overline{\text{DBG\_SS}}$ Lines form the SPI Debug Interface. These pins should be routed to test points on all designs, such that they can be connected to external hardware during system development. See also Section 13.1 on page 55.

Note that the $\overline{\text{DBG\_SS}}$ line shares the same pin as GPIO1. Only one of these two functions can be chosen and the circuit should be designed accordingly. The pull-up resistor in the schematics is optional and should be present only if the line is used as $\overline{\text{DBG\_SS}}$.

The DBG_CLK, DBG_DATA and $\overline{\text{DBG\_SS}}$ lines should not be connected to power or GND. For this reason, if these pins are totally unused (that is, they are not being used as debug or GPIO pins), they should be set as outputs. Refer to the GPIO/PWM Configuration T19 object in the *mXT640U 1.0 Protocol Guide* for more information.

4. Touchscreen Basics

4.1 Sensor Construction

A touchscreen is usually constructed from a number of transparent electrodes. These are typically on a glass or plastic substrate. They can also be made using non-transparent electrodes, such as copper or carbon. Electrodes are constructed from Indium Tin Oxide (ITO) or metal mesh. Thicker electrodes yield lower levels of resistance (perhaps tens to hundreds of Ω/square) at the expense of reduced optical clarity. Lower levels of resistance are generally more compatible with capacitive sensing. Thinner electrodes lead to higher levels of resistance (perhaps hundreds to thousands of Ω/square) with some of the best optical characteristics.

Interconnecting tracks can cause problems. The excessive RC time constants formed between the resistance of the track and the capacitance of the electrode to ground can inhibit the capacitive sensing function. In such cases, the tracks should be replaced by screen printed conductive inks (non-transparent) outside the touchscreen viewing area.

4.2 Electrode Configuration

The specific electrode designs used in Atmel touchscreens are the subject of various patents and patent applications. Further information is available on request.

The device supports various configurations of electrodes as summarized in Section 5. on page 18.

4.3 Scanning Sequence

All nodes are scanned in sequence by the device. There is a full parallelism in the scanning sequence to improve overall response time. The nodes are scanned by measuring capacitive changes at the intersections formed between the first X line and all the Y lines. Then the intersections between the next X line and all the Y lines are scanned, and so on, until all X and Y combinations have been measured.

The device can be configured in various ways. It is possible to disable some nodes so that they are not scanned at all. This can be used to improve overall scanning time.

4.4 Touchscreen Sensitivity

4.4.1 Adjustment

Sensitivity of touchscreens can vary across the extents of the electrode pattern due to natural differences in the parasitic capacitance of the interconnections, control chip, and so on. An important factor in the uniformity of sensitivity is the electrode design itself. It is a natural consequence of a touchscreen pattern that the edges form a discontinuity and hence tend to have a different sensitivity. The electrodes at the far edges do not have a neighboring electrode on one side and this affects the electric field distribution in that region.

A sensitivity adjustment is available for the whole touchscreen. This adjustment is a basic algorithmic threshold that defines when a node is considered to have enough signal change to qualify as being in detect.

4.4.2 Mechanical Stackup

The mechanical stackup refers to the arrangement of material layers that exist above and below a touchscreen. The arrangement of the touchscreen in relation to other parts of the mechanical stackup has an effect on the overall sensitivity of the screen. QMatrix technology has an excellent ability to operate in the presence of ground planes close to the sensor. QMatrix sensitivity is attributed more to the interaction of the electric fields between the transmitting (X) and receiving (Y) electrodes than to the surface area of these electrodes. For this reason, stray capacitance on the X or Y electrodes does not strongly reduce sensitivity.

Front panel dielectric material has a direct bearing on sensitivity. Plastic front panels are usually suitable up to about 1.2 mm, and glass up to about 2.5 mm (dependent upon the screen size and layout). The thicker the front panel, the lower the signal-to-noise ratio of the measured capacitive changes and hence the lower the resolution of the touchscreen. In general, glass front panels are near optimal because they conduct electric fields almost twice as easily as plastic panels.

Note: Care should be taken using ultra-thin glass panels as retransmission effects can occur, which can significantly degrade performance.

5. Sensor Layout

The specific electrode designs used in Atmel touchscreens are the subject of various patents and patent applications. Further information is available on request.

The physical matrix can be configured to have one or more touch objects. These are configured using the appropriate touch objects (Multiple Touch Touchscreen and Key Array). It is not mandatory to have all the allowable touch objects present. The objects are disabled by default so only those that you wish to use need to be enabled. Refer to the *mXT640U 1.0 Protocol Guide* for more information on configuring the touch objects.

The device supports various configurations of electrodes as summarized below:

- Touchscreen: 32 X × 20 Y maximum (subject to other configurations)
- Standard Keys: Up to 32 keys in an X/Y grid (Key Array), implemented using standard sense lines
- Generic Keys: Up to 3 keys in an X/Y grid (Key Array), implemented using the Generic Key lines

Note that the 3 nodes provided by the Generic Key lines are in addition to the maximum 640 nodes permitted on the device. Note also that the Key Array must contain either Generic Key lines or standard sense lines, but not both.

When designing the physical layout of the touch panel, obey the following rules:

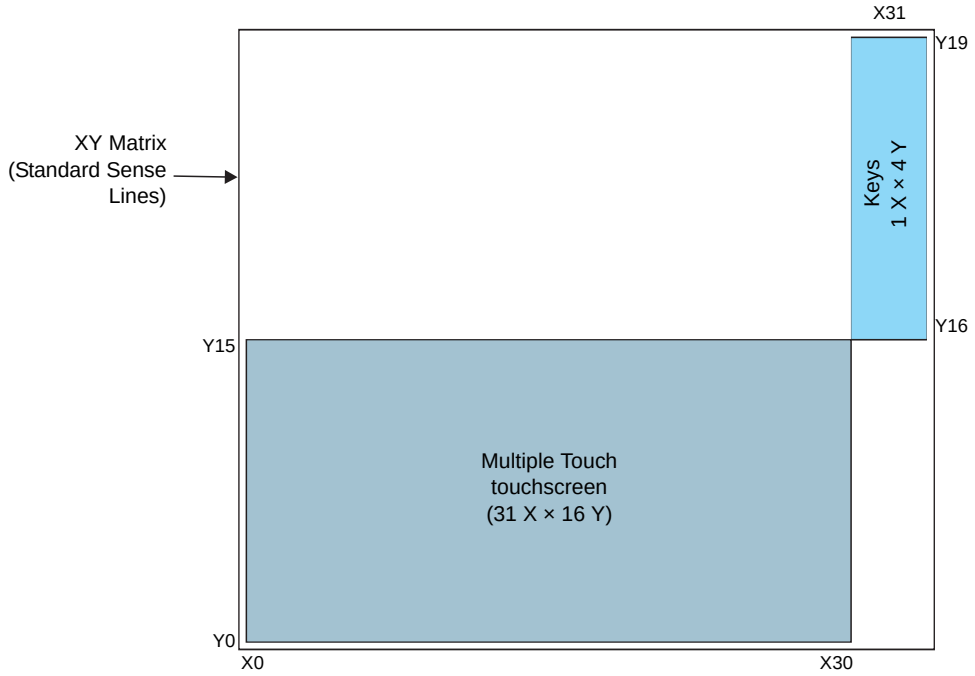
- Each touch object should be a regular rectangular shape in terms of the lines it uses.
- A Touchscreen object cannot share an X or Y line with another touch object if self-capacitance measurements are enabled.
- It is recommended that the Touchscreen should start at X0, Y0; if self-capacitance measurements are enabled, the Touchscreen **must** start at X0, Y0.
- Self Capacitance Touchscreens must have an even number of Y lines if low frequency compensation is used.
- It is recommended that a standard Key Array should occupy the highest X and Y lines.

5.1 Standard Key Array

For optimal performance in terms of cycle time overhead, it is recommended that the number of X (drive) lines used for the standard Key Array is kept to the minimum and designs should favor using Y lines where possible.

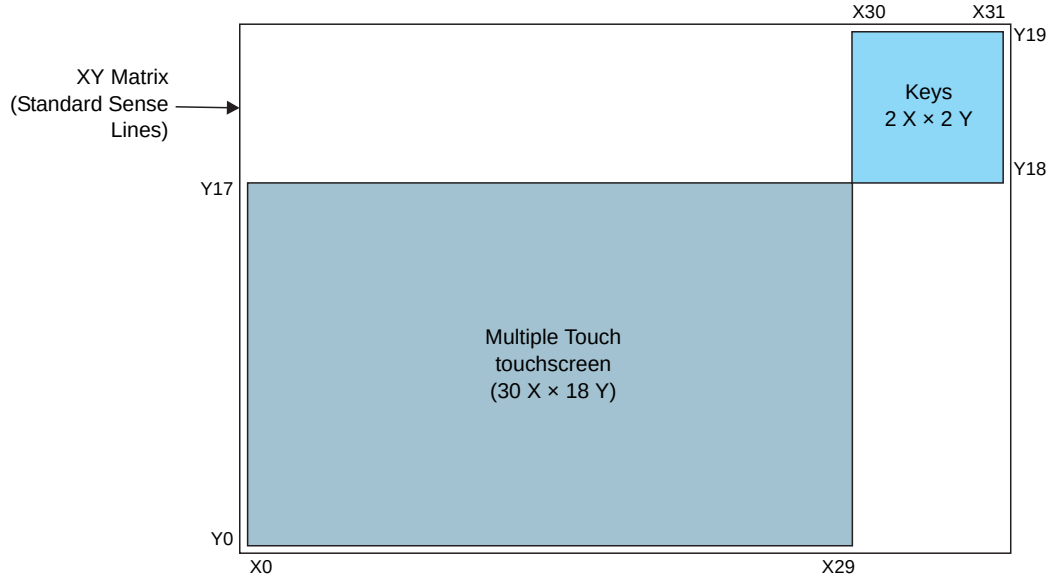
Figure 5-1 on page 19 shows an example layout for a Key Array of 1 X × 4 Y lines. Note that in this case using 1 X × 4 Y lines would give better performance than using 4 X × 1 Y lines.

Figure 5-1. Example Layout – Optimal Cycle Time



If, however, the intention is to preserve a larger touchscreen size and maintain an optimal aspect ratio, then using equal X and Y lines for the key array can be considered, as in Figure 5-2 on page 19.

Figure 5-2. Example Layout – Optimal Aspect Ratio

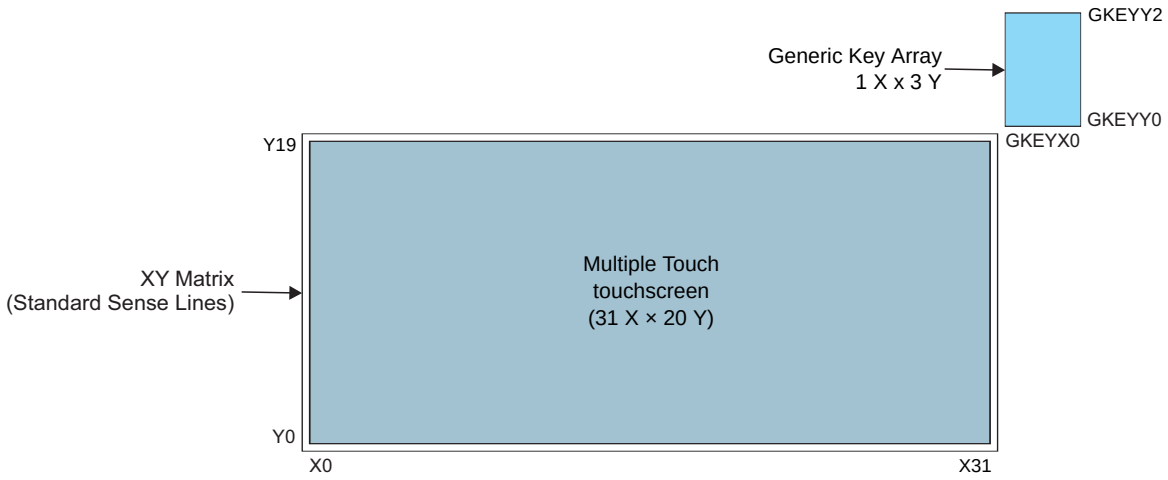


5.2 Generic Key Array

The Generic Key lines can be used to form an additional 3 mutual capacitance nodes that can be used to form a Key Array only.

Using the Generic Keys may add extra noise line measurements, which will impact power consumption and timings. It is therefore recommended that, where spare mutual capacitance sense lines are available, the sense lines are used to form a standard Key Array in preference to using the Generic Key lines.

Figure 5-3. Example Layout – Touchscreen with Generic Keys

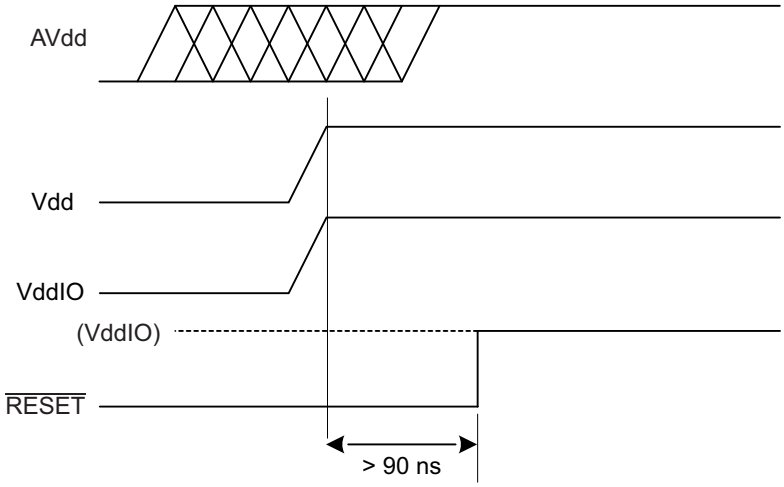


6. Power-up / Reset Requirements

There is an internal Power-on Reset (POR) in the device.

If an external reset is to be used the device must be held in $\overline{\text{RESET}}$ (active low) while the digital (Vdd) analog (AVdd) and I/O (VddIO) power supplies are powering up. The supplies must have reached their nominal values before the $\overline{\text{RESET}}$ signal is deasserted (that is, goes high). This is shown in Figure 6-1. See Section 14.2 on page 56 for nominal values for Vdd, VddIO, and AVdd.

Figure 6-1. Power Sequencing on the mXT640U



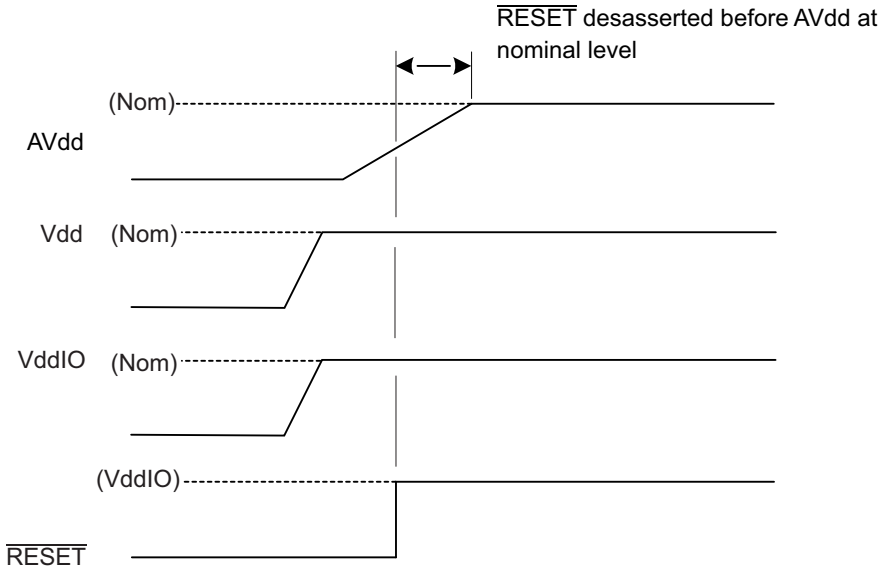
Note: When using external $\overline{\text{RESET}}$ at power-up, VddIO must not be enabled after Vdd

After power-up, the device typically takes 38 ms before it is ready to start communications.

If the $\overline{\text{RESET}}$ line is released before the AVdd supply has reached its nominal voltage (see Figure 6-2 on page 22), then some additional operations need to be carried out by the host. There are two options open to the host controller:

- Start the part in deep sleep mode and then send the command sequence to set the cycle time to wake the part and allow it to run normally. Note that in this case a calibration command is also needed.
- Send a reset command.

Figure 6-2. Power Sequencing on the mXT640U – Late rise on AVdd



The RESET pin can be used to reset the device whenever necessary. The RESET pin must be asserted low for at least 90 ns to cause a reset. After releasing the RESET pin the device typically takes 38 ms before it is ready to start communications. It is recommended to connect the RESET pin to a host controller to allow it to initiate a full hardware reset without requiring a power-down.

Make sure that any lines connected to the device are below or equal to Vdd during power-up. For example, if RESET is supplied from a different power domain to the VDDIO pin, make sure that it is held low when Vdd is off. If this is not done, the RESET signal could parasitically couple power via the RESET pin into the Vdd supply.

Note that the voltage level on the RESET pin of the device must never exceed VddIO (digital supply voltage).

A software reset command can be used to reset the chip (refer to the Command Processor T6 object in the *mXT640U 1.0 Protocol Guide*. A software reset takes typically 57 ms. After the chip has finished it asserts the CHG line to signal to the host that a message is available. The reset flag is set in the Message Processor object to indicate to the host that it has just completed a reset cycle. This bit can be used by the host to detect any unexpected brownout events. This allows the host to take any necessary corrective actions, such as reconfiguration.

A checksum check is performed on the configuration settings held in the nonvolatile memory. If the checksum does not match a stored copy of the last checksum, then this indicates that the settings have become corrupted. This is signaled to the host by setting the configuration error bit in the message data for the Command Processor T6 object (refer to the *mXT640U 1.0 Protocol Guide* for more information).

Note that the CHG line is briefly set as an input during power-up or reset. It is therefore particularly important that the line should be allowed to float high via the CHG line pull-up resistor during this period. It should not be driven by the host (see Table 14.7.3 on page 63).

At power-on, the device performs a self-test routine to check for shorts that might cause damage to the device. Refer to the Self Test T25 object in the *mXT640U 1.0 Protocol Guide* for more details about this process.

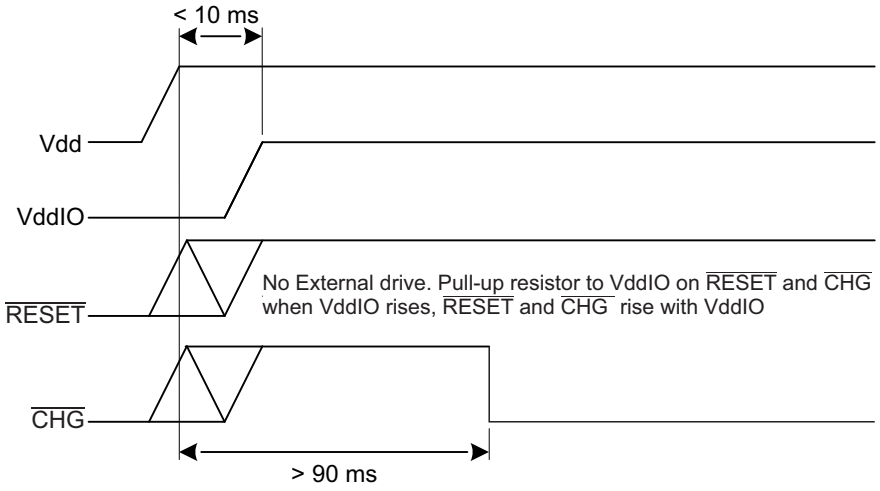
Warning: The device should only be reset by using the RESET line. If an attempt is made to reset by removing the power from the device it will not reset correctly as power will be drawn from the interface lines.

6.1 Power-up and Reset Sequence – VddIO Enabled after Vdd

The Power-up sequence that can be used in applications where VddIO must be powered up after Vdd, is shown in Figure 6-3.

In this case the communication interface to the maXTouch device is not driven by the host system. The $\overline{\text{RESET}}$ and $\overline{\text{CHG}}$ pins are connected to VddIO using suitable pull-up resistors. Vdd is powered up, followed by VddIO, no more than 10 ms after Vdd. Due to the pull-up resistors, $\overline{\text{RESET}}$ and $\overline{\text{CHG}}$ will rise with VddIO. The internal POR system ensures reliable boot up of the device and the $\overline{\text{CHG}}$ line will go low approximately 38 ms after Vdd to notify the host that the device is ready to start communication.

Figure 6-3. Power-up Sequence



6.1.1 Summary

The Power-up and reset requirements for the maXTouch devices are summarized in Table 6-1.

Table 6-1. Power-up and Reset Requirements

Condition	External $\overline{\text{RESET}}$	VddIO Delay (After Vdd)	AVdd Power-Up	Comments
1	Low at Power-up	0 ms	Before $\overline{\text{RESET}}$ is released	If AVdd bring-up is delayed then additional actions will be required by the host. See notes in Figure 6-1 on page 21
2	Not driven	<10 ms	Before VddIO	

7. Detailed Operation

7.1 Touch Detection

The mXT640U allows for both mutual and self capacitance measurements, with the self capacitance measurements being used to augment the mutual capacitance measurements to produce reliable touch information.

When self capacitance measurements are enabled, touch classification is achieved using both mutual and self capacitance touch data. This has the advantage that both types of measurement systems can work together to detect touches under a wide variety of circumstances.

Mutual capacitance touch data is used wherever possible to classify touches as this has greater granularity than self capacitance measurements and provides positional information on touches. Refer to the *mXT640U 1.0 Protocol Guide* for more information on measurements.

Self capacitance measurements, on the other hand, allow for the detection of single touches in extreme case, such as single thick glove touches, when touches can only be detected by self capacitance data and may be missed by mutual capacitance touch detection.

7.2 Operational Modes

The device operates in two modes: Active (touch detected) and Idle (no touches detected). Both modes operate as a series of burst cycles. Each cycle consists of a short burst (during which measurements are taken) followed by an inactive sleep period. The difference between these modes is the length of the cycles. Those in idle mode typically have longer sleep periods. The cycle length is configured using the IDLEACQINT and ACTVACQINT settings in the Power Configuration T7. In addition, an *Active to Idle Timeout* setting is provided.

Refer to the *mXT640U 1.0 Protocol Guide* for full information on how these modes operate, and how to use the settings provided.

7.3 Detection Integrator

The device features a touch detection integration mechanism. This acts to confirm a detection in a robust fashion. A counter is incremented each time a touch has exceeded its threshold and has remained above the threshold for the current acquisition. When this counter reaches a preset limit the sensor is finally declared to be touched. If, on any acquisition, the signal is not seen to exceed the threshold level, the counter is cleared and the process has to start from the beginning.

The detection integrator is configured using the appropriate touch objects (Multiple Touch Touchscreen T100, Key Array T15). Refer to the *mXT640U 1.0 Protocol Guide* for more information.

7.4 Sensor Acquisition

The maximum acquisition time for one X line on the mXT640U is 5 µs. Care should be taken to ensure that the total time for one X line configured by the Acquisition Configuration T8 and CTE Configuration T46 objects do not exceed this (refer to the *mXT640U 1.0 Protocol Guide* for details on these objects).

7.5 Calibration

Calibration is the process by which a sensor chip assesses the background capacitance on each node. Nodes are only calibrated on reset and when:

- The node is enabled (that is, activated).

or

- The node is already enabled and one of the following applies:
  - The node is held in detect for longer than the Touch Automatic Calibration setting (refer to the *mXT640U 1.0 Protocol Guide* for more information on TCHAUTOCAL setting in the Acquisition Configuration object).

- The signal delta on a node is at least the touch threshold (TCHTHR – TCHHYST) in the anti-touch direction, while it meets the criteria in the Touch Recovery Processes that results in a recalibration. (Refer to the *mXT640U 1.0 Protocol Guide* for objects Acquisition Configuration T8 and Self Capacitance Configuration T111).
- The host issues a recalibrate command.
- Certain configuration settings are changed.

A status message is generated on the start and completion of a calibration.

Note that the device performs a global calibration; that is, all the nodes are calibrated together.

7.6 Digital Filtering and Noise Suppression

The mXT640U supports on-chip filtering of the acquisition data received from the sensor. Specifically, the maXCharger T72 object provides an algorithm to suppress the effects of noise (for example, from a noisy charger plugged into the user's product). This algorithm can automatically adjust some of the acquisition parameters on-the-fly to filter the analog-to-digital conversions (ADCs) received from the sensor.

Additional noise suppression is provided by the Self Capacitance maXCharger T108 object. Similar in both design and configuration to the maXCharger T72 object, the Self Capacitance maXCharger T108 object is the noise suppression interface for self capacitance touch measurements.

Noise suppression is triggered when a noise source is detected.

- A hardware trigger can be implemented using the NOISE_IN pin.
- The host driver code can indicate when a noise source is present.
- The noise suppression is also triggered based on the noise levels detected using internal line measurements.The maXCharger T72 and Self Capacitance maXCharger T108 object selects the appropriate controls to suppress the noise present in the system.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the maXCharger T72 and Self Capacitance maXCharger T108 objects.

7.7 Shieldless Support and Display Noise Suppression

The mXT640U can support shieldless sensor design even with a noisy LCD by using the following features.

- **Optimal Integration:** This feature is not filtering as such, but enables the user to use a shorter integration window. The integration window optimizes the amount of charge collected against the amount of noise collected, to ensure an optimal SNR. This feature also benefits the system in the presence of an external noise source. This feature is configured using the Shieldless T56 object. Refer to the *mXT640U 1.0 Protocol Guide* for more information
- **Display noise suppression:** This feature is based on filtering provided by the Lens Bending T65 object (See Section 7.10 on page 26). This feature allows the device to overcome display noise simultaneously with external noise. Refer to the *mXT640U 1.0 Protocol Guide* for more information

7.8 Retransmission Compensation

The device can limit the undesirable effects on the mutual capacitance touch signals caused by poor device coupling to ground, such as poor sensitivity and touch break-up. This is achieved using the Retransmission Compensation T80 object. This object can be configured to allow the touchscreen to compensate for signal degradation due to these undesirable effects. If self capacitance measurements are also scheduled, the Retransmission Compensation T80 object will use the resultant data to enhance the compensation process.

The Retransmission Compensation T80 object is also capable of compensating for water presence on the sensor if self capacitance measurements are scheduled. In this case, both mutual capacitance and self capacitance measurements are used to detect moisture and then, once moisture is detected, self capacitance measurements are used to detect single touches in the presence of moisture.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the Retransmission Compensation T80 object.

7.9 Grip Suppression

The device has two grip suppression mechanisms to suppress false detections from a user's grip.

Mutual grip suppression works by specifying a boundary around a touchscreen, within which touches can be suppressed whilst still allowing touches in the center of the touchscreen. This ensures that a "rolling" hand touch (such as when a user grips a mobile device) is suppressed. A "real" (finger) touch towards the center of the screen is allowed.

Mutual grip suppression is configured using the Grip Suppression T40 object. There is one instance of the Grip Suppression T40 object for each Multiple Touch Touchscreen T100 object present on the device.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the Grip Suppression T40 object.

Self Capacitance grip suppression works by looking for characteristic shapes in the self capacitance measurement along the touchscreen boundary, and thereby distinguishing between a grip and a touch further into the sensor.

7.10 Lens Bending

The device supports algorithms to eliminate disturbances from the measured signal.

When the sensor suffers from the screen deformation (lens bending) the signal values acquired by normal procedure are corrupted by the disturbance component (bend). The amount of bend depends on:

- The mechanical and electrical characteristics of the sensor
- The amount and location of the force applied by the user touch to the sensor

The Lens Bending T65 object measures the bend component and compensates for any distortion caused by the bend. As the bend component is primarily influenced by the user touch force, it can be used as a secondary source to identify the presence of a touch. The additional benefit of the Lens Bending T65 object is that it will eliminate LCD noise as well.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the Lens Bending T65 object.

7.11 Glove Detection

The device has glove detection algorithms that process the measurement data received from the touchscreen classifying touches as potential gloved touches.

The Glove Detection T78 object is used to detect glove touches. In Normal Mode the Glove Detection T78 object applies vigorous glove classification to small signal touches to minimize the effect of unintentional hovering finger reporting. Once a gloved touch is found, the Glove Detection T78 object enters Glove Confidence Mode. In this mode the device expects the user to be wearing gloves so the classification process is much less stringent.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the Glove Detection T78 object.

7.12 Stylus Support

The mXT640U allows for the particular characteristics of passive stylus touches, whilst still allowing conventional finger touches to be detected. The touch sensitivity and threshold controls for stylus touches are configured separately from those for conventional finger touches so that both types of touches can be accommodated.

Stylus support ensures that the small touch area of a stylus registers as a touch, as this would otherwise be considered too small for the touchscreen. Additionally, there are controls to distinguish a stylus touch from an unwanted approaching finger (such as on the hand holding the stylus).

Passive stylus touches are configured by the Passive Stylus T47 object. There is one instance of the Passive Stylus T47 object for each Multiple Touch Touchscreen T100 object present on the device.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on configuring a stylus.

7.13 Unintentional Touch Suppression

The Touch Suppression T42 object provides a mechanism to suppress false detections from unintentional touches from a large body area, such as from a face, ear or palm. The Touch Suppression T42 object also provides Maximum Touch Suppression to suppress all touches if more than a specified number of touches has been detected. There is one instance of the Touch Suppression T42 object for each Multiple Touch Touchscreen T100 object present on the device.

7.14 Adjacent Key Suppression Technology

Adjacent Key Suppression (AKS) technology is a patented method used to detect which touch object is touched when objects are located close together. A touch in a group of AKS objects is only indicated on the object in that group that is touched first. This is assumed to be the intended object. Once an object in an AKS group is in detect, there can be no further detections within that group until the object is released. Objects can be in more than one AKS group.

Note that AKS technology works best when it operates in conjunction with a detect integration setting of several acquisition cycles.

The device has two levels of AKS. The first level works between the touch objects (Multiple Touch Touchscreen T100 and Key Array T15). The touch objects are assigned to AKS groups. If a touch occurs within one of the touch objects in a group, then touches within other objects inside that group are suppressed. For example, if a touchscreen and a Key Array are placed in the same AKS group, then a touch in the touchscreen will suppress touches in the Key Array, and vice versa.

The second level of AKS is internal AKS within an individual Key Array object (note that internal AKS is not present on other types of touch objects, only a Key Array T15). If internal AKS is enabled, then when one key is touched, touches on all the other keys within the Key Array are suppressed.

AKS is configured using the touch objects (Multiple Touch Touchscreen T100 or Key Array T15).

Refer to the *mXT640U 1.0 Protocol Guide* for more information.

Note: If a touch is in detect and then AKS is enabled, that touch will not be forced out of detect. It will not go out of detect until the touch is released. AKS will then operate normally. This applies to both levels of AKS.

8. Host Communications

Communication with the host is achieved using one of the following interfaces:

- I²C (see Section 9. on page 30)
- HID-I²C (see Section 10. on page 36)

Any interface can be used, depending on the needs of the user's project, but only one interface should be used in any one design.

8.1 I²C Mode Selection (I2CMODE Pin)

The selection of the I²C or the HID-I²C mode is determined by connecting the I2CMODE pin according to Table 8-1.

Table 8-1. I²C Mode Selection

I2CMODE	Interface Selected
Connected to GND	HID-I²C
Pulled up to VddIO	I²C
Floating	Mode is selected according to the I²C address (as determined by the ADDSEL pin). See Section 8.1.1 for more information.

8.1.1 Automatic Selection of I²C and HID-I²C Modes

If the I2CMODE pin is left floating (that is, automatic mode selection), the device will listen on both I²C addresses and automatically select the protocol to be used depending on the first message received. In this case the ADDSEL pin determines the primary and secondary I²C addresses, and these in turn determine the communications mode to be used. If the primary I²C address is detected, I²C is used for communications; if the I²C secondary address is detected, HID-I²C is used.

The selection of both the communications mode and the I²C addresses is summarized in Table 8-2 on page 28.

Table 8-2. Communications Mode Selection

I2CMODE	ADDSEL	Mode
0 (HID-I²C selected)	0 (Address = 0x4A)	HID-I²C communications at 0x4A
	1 (Address = 0x4B)	HID-I²C communications at 0x4B
1 (I²C selected)	0 (Address = 0x4A)	I²C communications at 0x4A
	1 (Address = 0x4B)	I²C communications at 0x4B
No input or input floating (auto selection)	0 (Primary address = 0x4A, secondary address = 0x4B)	I²C communications at 0x4A (primary address) HID-I²C communications at 0x4B (secondary address)
	1 (Primary address = 0x4B, secondary address = 0x4A)	I²C communications at 0x4B (primary address) HID-I²C communications at 0x4A (secondary address)

8.2 I²C Address Selection (ADDSEL Pin)

If the I2CMODE pin is not floating (that is, a particular mode is chosen), the I²C address is selected by connecting the ADDSEL pin according to Table 8-3.

Table 8-3. I²C Address Selection

ADDSEL	I²C Address
Connected to GND	0x4A
Pulled up to VddIO	0x4B

9. I²C Communications

The device can use an I²C interface for communication.

The I²C interface is used in conjunction with the $\overline{CHG}$ line. The $\overline{CHG}$ line going active signifies that a new data packet is available. This provides an interrupt-style interface and allows the device to present data packets when internal changes have occurred.

See Section 8. on page 28 for information on selecting I²C mode.

9.1 I²C Addresses

The device supports two I²C device addresses that are selected using the ADDSEL line at start up. The two internal I²C device addresses are 0x4A and 0x4B. The selection of the address (and the communication mode) is described in Section 8.2 on page 29.

The I²C address is shifted left to form the SLA+W or SLA+R address when transmitted over the I²C interface, as shown in Table 9-1.

Table 9-1. Format of an I²C Address

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0x4A or 0x4B							Read/write

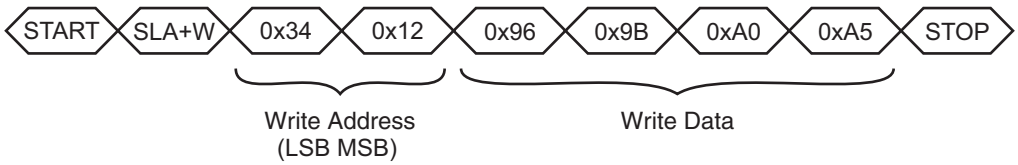
9.2 Writing To the Device

A WRITE cycle to the device consists of a START condition followed by the I²C address of the device (SLA+W). The next two bytes are the address of the location into which the writing starts. The first byte is the Least Significant Byte (LSByte) of the address, and the second byte is the Most Significant Byte (MSByte). This address is then stored as the address pointer.

Subsequent bytes in a multi-byte transfer form the actual data. These are written to the location of the address pointer, location of the address pointer + 1, location of the address pointer + 2, and so on. The address pointer returns to its starting value when the WRITE cycle STOP condition is detected.

Figure 9-1 shows an example of writing four bytes of data to contiguous addresses starting at 0x1234.

Figure 9-1. Example of a Four-byte Write Starting at Address 0x1234

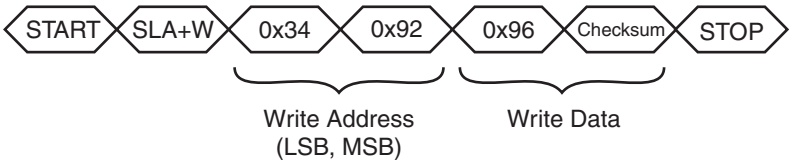


9.3 I²C Writes in Checksum Mode

In I²C checksum mode an 8-bit CRC is added to all I²C writes. The CRC is sent at the end of the data write as the last byte before the STOP condition. All the bytes sent are included in the CRC, including the two address bytes. Any command or data sent to the device is processed even if the CRC fails.

To indicate that a checksum is to be sent in the write, the most significant bit of the MSByte of the address is set to 1. For example, the I²C command shown in Figure 9-2 writes a value of 150 (0x96) to address 0x1234 with a checksum. The address is changed to 0x**9**234 to indicate checksum mode.

Figure 9-2. Example of a Write To Address 0x1234 With a Checksum



9.4 Reading From the Device

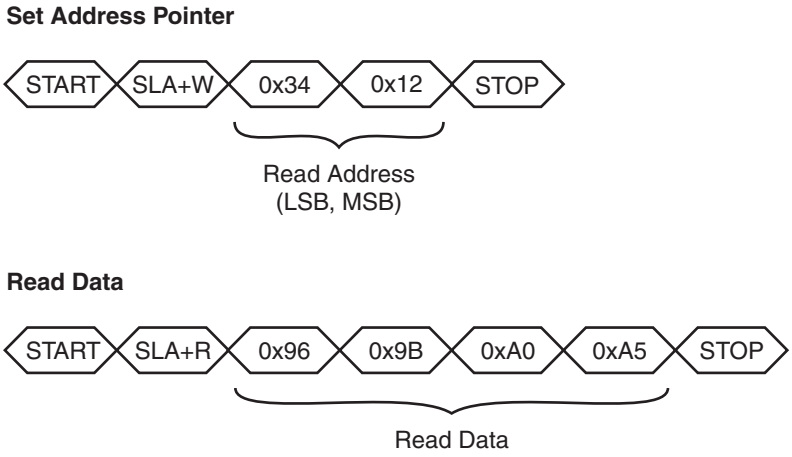
Two I²C bus activities must take place to read from the device. The first activity is an I²C write to set the address pointer (LSByte then MSByte). The second activity is the actual I²C read to receive the data. The address pointer returns to its starting value when the read cycle NACK is detected.

It is not necessary to set the address pointer before every read. The address pointer is updated automatically after every read operation. The address pointer will be correct if the reads occur in order. In particular, when reading multiple messages from the Message Processor T5 object, the address pointer is automatically reset to allow continuous reads (see Section 9.5).

The WRITE and READ cycles consist of a START condition followed by the I²C address of the device (SLA+W or SLA+R respectively). Note that in this mode, calculating a checksum of the data packets is not supported.

Figure 9-3 shows the I²C commands to read four bytes starting at address 0x1234.

Figure 9-3. Example of a Four-byte Read Starting at Address 0x1234



9.5 Reading Status Messages with DMA

The device facilitates the easy reading of multiple messages using a single continuous read operation. This allows the host hardware to use a direct memory access (DMA) controller for the fast reading of messages, as follows:

1. The host uses a write operation to set the address pointer to the start of the Message Count T44 object, if necessary (1). If a checksum is required on each message, the most significant bit of the MSByte of the read address must be set to 1.
2. The host starts the read operation of the message by sending a START condition.
3. The host reads the Message Count T44 object (one byte) to retrieve a count of the pending messages (refer to the *mXT640U 1.0 Protocol Guide* for details).
4. The host calculates the number of bytes to read by multiplying the message count by the size of the Message Processor T5 object (2).

1. The STOP condition at the end of the read resets the address pointer to its initial location, so it may already be pointing at the Message Count T44 object following a previous message read.
2. The host should have already read the size of the Message Processor T5 object in its initialization code.

5. Note that the size of the Message Processor T5 object as recorded in the Object Table includes a checksum byte. If a checksum has not been requested, one byte should be deducted from the size of the object. That is: number of bytes = count × (size – 1).
6. The host reads the calculated number of message bytes. It is important that the host does *not* send a STOP condition during the message reads, as this will terminate the continuous read operation and reset the address pointer. No START and STOP conditions must be sent between the messages.
7. The host sends a STOP condition at the end of the read operation after the last message has been read. The NACK condition immediately before the STOP condition resets the address pointer to the start of Message Count T44 object.

Figure 9-4 shows an example of using a continuous read operation to read three messages from the device without a checksum. Figure 9-5 on page 33 shows the same example with a checksum.

Figure 9-4. Continuous Message Read Example – No Checksum

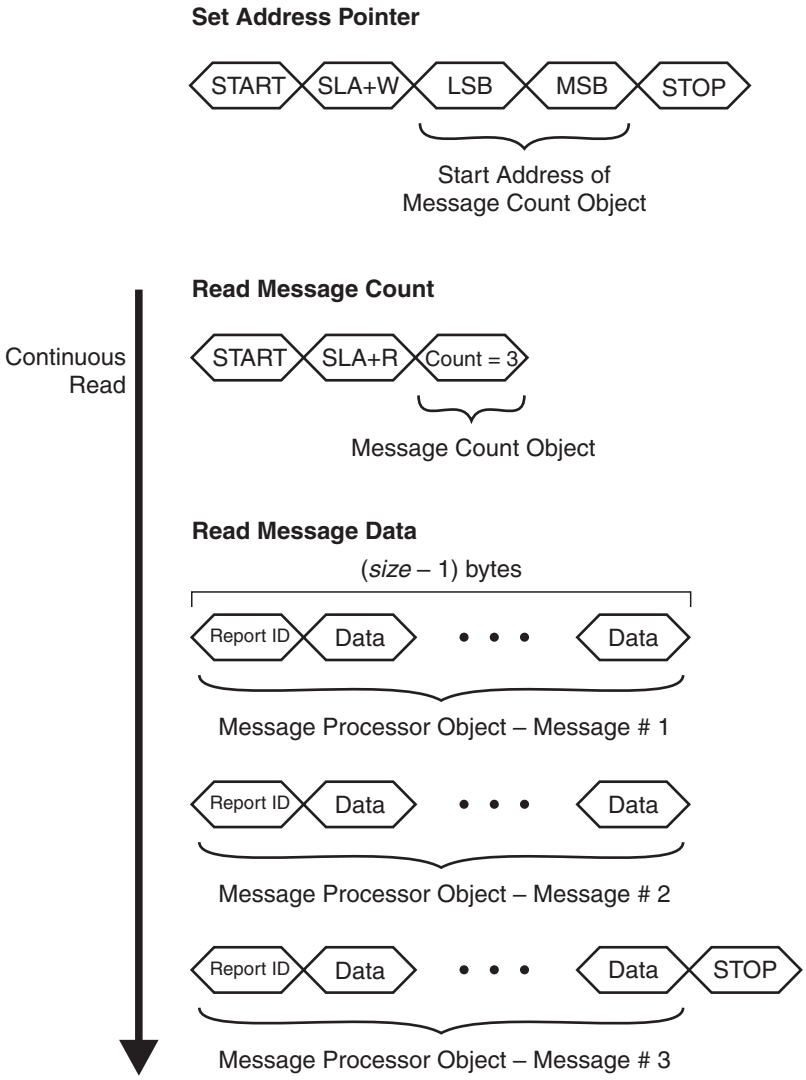
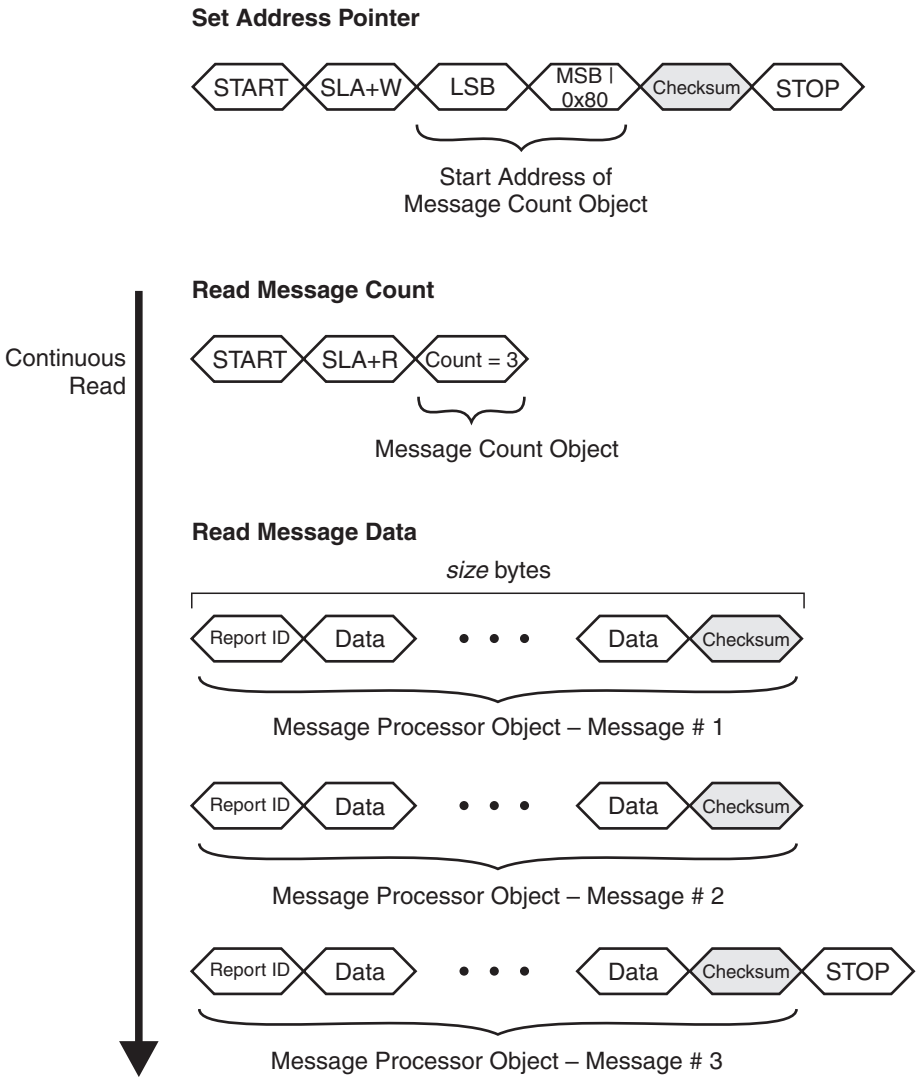


Figure 9-5. Continuous Message Read Example – I²C Checksum Mode



There are no checksums added on any other I²C reads. An 8-bit CRC can be added, however, to all I²C writes, as described in Section 9.3 on page 30.

An alternative method of reading messages using the $\overline{\text{CHG}}$ line is given in Section 9.6.

9.6 $\overline{\text{CHG}}$ Line

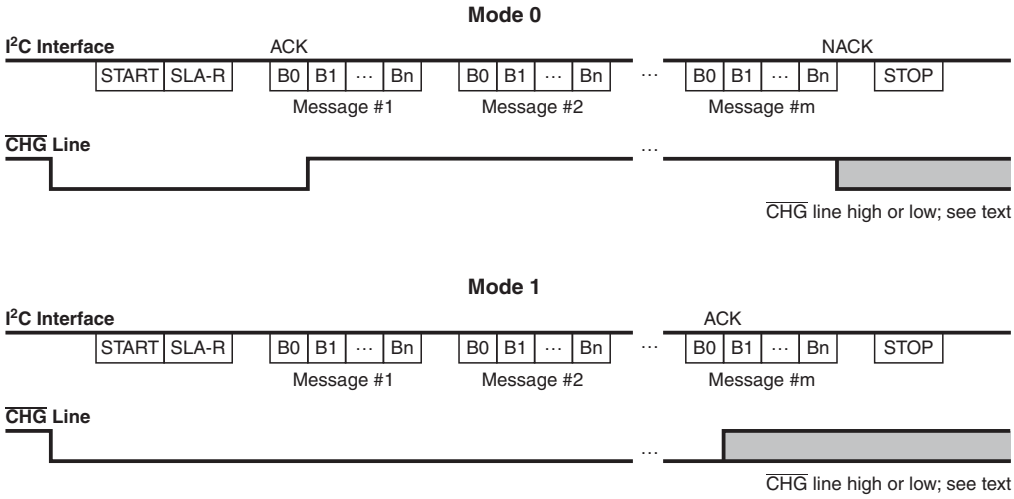
The $\overline{\text{CHG}}$ line is an active-low, open-drain output that is used to alert the host that a new message is available in the Message Processor T5 object. This provides the host with an interrupt-style interface with the potential for fast response times. It reduces the need for wasteful I²C communications.

The $\overline{\text{CHG}}$ line should always be configured as an input on the host during normal usage. This is particularly important after power-up or reset (see Section 6. on page 21).

A pull-up resistor is required, typically 3.3 kΩ to VddIO.

The $\overline{\text{CHG}}$ line operates in two modes, as defined by the Communications Configuration T18 object (refer to the *mXT640U 1.0 Protocol Guide*).

Figure 9-6. $\overline{\text{CHG}}$ Line Modes for I²C-compatible Transfers



In Mode 0 (edge-triggered operation):

1. The $\overline{\text{CHG}}$ line goes low to indicate that a message is present.
2. The $\overline{\text{CHG}}$ line goes high when the first byte of the first message (that is, its report ID) has been sent and acknowledged (ACK sent) and the next byte has been prepared in the buffer.
3. The STOP condition at the end of an I²C transfer causes the $\overline{\text{CHG}}$ line to stay high if there are no more messages. Otherwise the $\overline{\text{CHG}}$ line goes low to indicate a further message.

Note that Mode 0 also allows the host to continually read messages by simply continuing to read bytes back without issuing a STOP condition. Message reading should end when a report ID of 255 (“invalid message”) is received. Alternatively the host ends the transfer by sending a NACK after receiving the last byte of a message, followed by a STOP condition. If there is another message present, the $\overline{\text{CHG}}$ line goes low again, as in step 1. In this mode the state of the $\overline{\text{CHG}}$ line does not need to be checked during the I²C read.

In Mode 1 (level-triggered operation):

1. The $\overline{\text{CHG}}$ line goes low to indicate that a message is present.
2. The $\overline{\text{CHG}}$ line remains low while there are further messages to be sent after the current message.
3. The $\overline{\text{CHG}}$ line goes high again only once the first byte of the last message (that is, its report ID) has been sent and acknowledged (ACK sent) and the next byte has been prepared in the output buffer.

Mode 1 allows the host to continually read the messages until the $\overline{\text{CHG}}$ line goes high, and the state of the $\overline{\text{CHG}}$ line determines whether or not the host should continue receiving messages from the device.

Note: The state of the $\overline{\text{CHG}}$ line should be checked only between messages and not between the bytes of a message. The precise point at which the $\overline{\text{CHG}}$ line changes state cannot be predicted and so the state of the $\overline{\text{CHG}}$ line cannot be guaranteed between bytes.

The Communications Configuration T18 object can be used to configure the behavior of the $\overline{\text{CHG}}$ line. In addition to the $\overline{\text{CHG}}$ line operation modes described above, this object allows direct control over the state of the $\overline{\text{CHG}}$ line. Refer to the *mXT640U 1.0 Protocol Guide* for more information.

9.7 SDA, SCL

The I²C bus transmits data and clock with SDA and SCL, respectively. These are open-drain. The device can only drive these lines low or leave them open. The termination resistors (Rp) pull the line up to VddIO if no I²C device is pulling it down.

The termination resistors should be chosen so that the rise times on SDA and SCL meet the I²C specifications for the interface speed being used, bearing in mind other loads on the bus (see Section 14.9 on page 64). For best latency performance, it is recommended that no other devices share the I²C bus with the maXTouch controller.

9.8 Clock Stretching

The device supports clock stretching in accordance with the I^2C specification. It may also instigate a clock stretch if a communications event happens during a period when the device is busy internally. The maximum clock stretch is approximately 10 – 15 ms.

10. HID-I²C Communications

The device is an HID-I²C device presenting two Top-level Collections (TLCs):

- **Generic HID-I²C** – Provides a generic HID-I²C interface that allows the host to communicate with the device using the object-based protocol (OBP).
- **Digitizer HID-I²C** – Supplies touch information to the host. This interface is supported by Microsoft Windows without the need for additional software.

See Section 8. on page 28 for information on selecting HID-I²C mode.

Other features are identical to standard I²C communication described in Section 9. on page 30.

Refer to the Microsoft HID-I²C documentation, *HID Over I²C Protocol Specification – Device Side*, for information on the HID-I²C specification.

10.1 I²C Addresses

See Section 9.1 on page 30.

10.2 Device

The device is compliant with HID-I²C specification V1.0. It has the following specification:

Vendor ID:	0x03EB (Atmel)
Product ID:	0x2163 (mXT640U)
Version:	16-bit Version & Build Identifier in the form 0xVVBB, where: VV = Version Major (Upper 4 bits) / Minor (Lower 4 bits) BB = Build number in BCD format
HID descriptor address:	0x0000

10.3 HID Descriptor

The host should read the HID descriptor on initialization to ascertain the key attribute of the HID device. These include the report description and the report ID to be used for communication with the HID device. The HID descriptor address is 0x0000.

Note that the host driver must not make any assumptions about the report packet formats, data locations or report IDs. These must be read from the HID descriptor as they may change in future versions of the firmware.

For more information on how to read the HID descriptor, refer to the Microsoft HID-I²C documentation.

10.4 HID-I²C Report IDs

Table 10-1 describes the HID-I²C report IDs used in reports sent to the host.

Table 10-1. HID-I²C Report IDs

Report ID	Description	Top-level Collection
0x06	Object Protocol (OBP) command and response (see Section 10.5 on page 37)	Generic HID-I²C
0x01	Touch report (see Section 10.6.1 on page 42)	Digitizer HID-I²C
0x02	Maximum Touches (Surface Contacts) report (see Section 10.6.3 on page 44)	Digitizer HID-I²C
0x05	Touch Hardware Quality Assurance (THQA) report (see Section 10.6.4 on page 44)	Digitizer HID-I²C

10.5 Generic HID-I²C

The Generic HID-I²C TLC supports an input report for receiving data from the device and an output report for sending data to the device.

Commands are sent by the host using the output reports. Responses from the device are sent using input reports.

Supported commands are:

- Read/Write Memory Map
- Send Auto-return Messages

The HID-I²C Report ID used is that for Object Protocol commands and responses; see Table 10-1 on page 36 for the value.

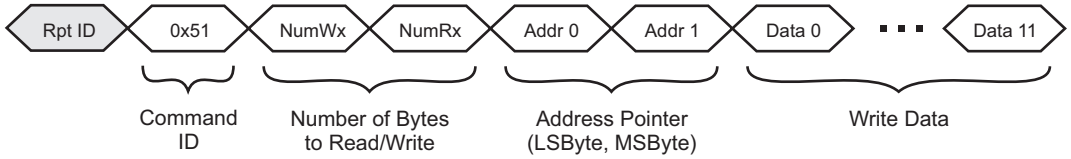
10.5.1 Read/Write Memory Map

10.5.1.1 Introduction

This command is used to carry out a write/read operation on the memory map of the device.

The command packet has the generic format given in Figure 10-1. The following sections give examples on using the command to write to the memory map and to read from the memory map.

Figure 10-1. Generic Command Packet Format

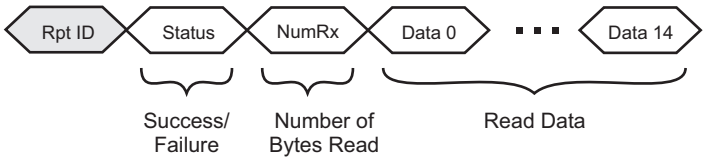


In Figure 10-1:

- **Rpt ID** is the HID-I²**C** Report ID used for Object Protocol commands and responses (see Table 10-1 on page 36).
- **NumWx** is the number of data bytes to write to the memory map (may be zero). If the address pointer is being sent, this must include the size of the address pointer.
- **NumRx** is the number of data bytes to read from the memory map (may be zero).
- **Addr 0** and **Addr 1** form the address pointer to the memory map (where necessary; may be zero if not needed).
- **Data 0** to **Data 11** are the bytes of data to be written (in the case of a write). Note that data locations beyond the number specified by NumWx will be ignored.

The response packet has the generic format given in Figure 10-2.

Figure 10-2. Response Packet Format



In Figure 10-2:

- **Rpt ID** is the HID-I²**C** Report ID used for Object Protocol commands and responses (see Table 10-1 on page 36).
- **Status** indicates the result of the command:
  - 0x00 = read and write completed; read data returned
  - 0x04 = write completed; no read data requested
- **NumRx** is the number of bytes following that have been read from the memory map (in the case of a read). This will be the same value as NumRx in the command packet.
- **Data 0** to **Data 14** are the data bytes read from the memory map.

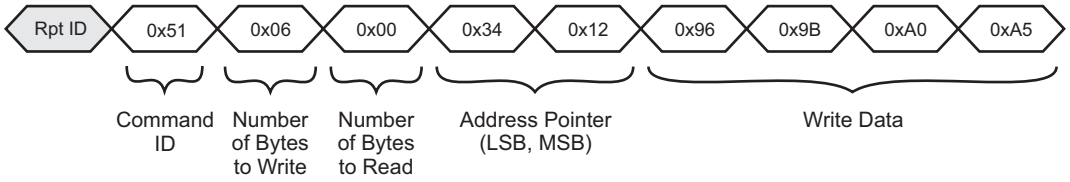
10.5.1.2 Writing To the Device

A write operation cycle to the device consists of sending a packet that contains six header bytes. These specify the HID-I^{2}C report ID, the Command ID, the number of bytes to read, the number of bytes to write, and the 16-bit address pointer.

Subsequent bytes in a multi-byte transfer form the actual data. These are written to the location of the address pointer, location of the address pointer +1, location of the address pointer + 2, and so on.

Figure 10-3 shows an example command packet to write four bytes of data to contiguous addresses starting at 0x1234.

Figure 10-3. Example of a Four-byte Write Starting at Address 0x1234

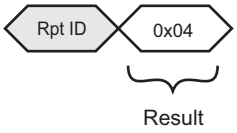


In Figure 10-3:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 10-1 on page 36).
- The number of bytes to read is set to zero as this is a write-only operation.
- The number of bytes to write is six: that is, four data bytes plus the two address pointer bytes.

Figure 10-4 shows the response to this command. Note that the result status returned is 0x04 (that is, the write operation was completed but no read data was requested). Note also that the Report ID is the same one used in the command packet.

Figure 10-4. Response to Example Four-byte Write

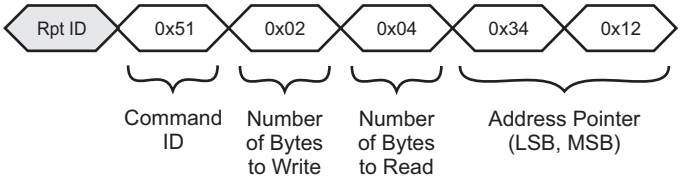


10.5.1.3 Reading From the Device

A read operation consists of sending a packet that contains the six header bytes only and no write data.

Figure 10-5 shows an example command packet to read four bytes starting at address 0x1234. Note that the address pointer is included in the number of bytes to write, so the number of bytes to write is set to 2 as there are no other data bytes to be written.

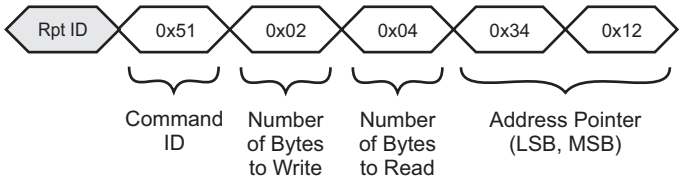
Figure 10-5. Example of a Four-byte Read Starting at Address 0x1234



It is not necessary to set the address pointer before every read. The address pointer is updated automatically after every read operation, so the address pointer will be correct if the reads occur in order.

Figure 10-6 shows the response to this command. The result status returned is 0x00 (that is the write operation was completed and the data was returned). The number of bytes returned will be the same as the number requested (4 in this case).

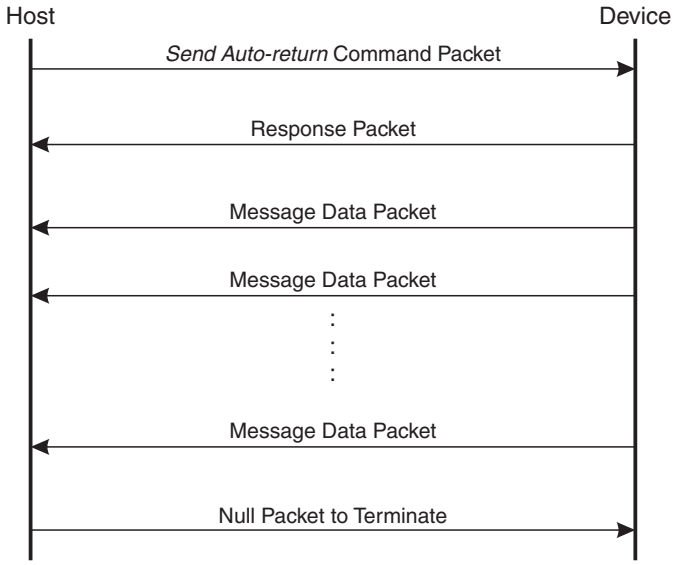
Figure 10-6. Response to Example Four-byte Read



10.5.2 Send Auto-return Messages

With this command the device can be configured to return new messages from the Message Processor T5 object autonomously. The packet sequence to do this is shown in Figure 10-7.

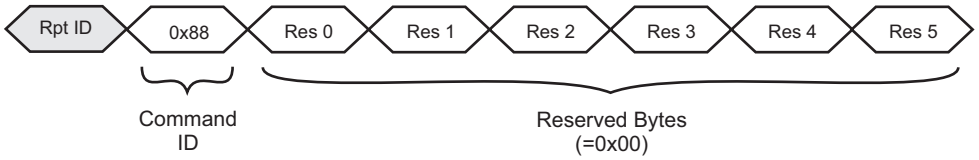
Figure 10-7. Packet Sequence for "Send Auto-return" Command



The HID-I^{2}C Report ID used is that for Object Protocol commands and responses; see Table 10-1 on page 36 for the value.

The command packet has the format given in Figure 10-8.

Figure 10-8. Command Packet Format

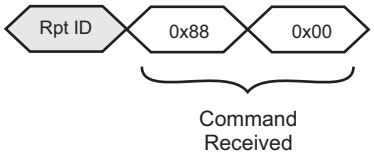


In Figure 10-8:

- **Rpt ID** is the HID-I^2**C** Report ID used for Object Protocol commands and responses (see Table 10-1 on page 36).
- **Res 0** to **Res 5** are reserved bytes with a value of 0x00.

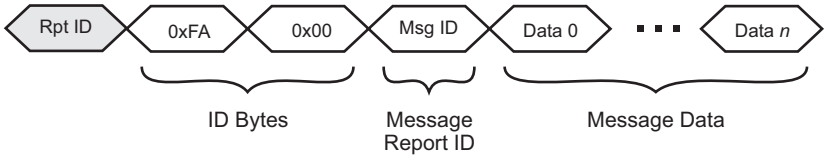
The response packet has the format given in Figure 10-9. Note that with this command, the command packet does not include an address pointer as the device already knows the address of the Message Processor T5 object.

Figure 10-9.Response Packet Format



Once the device has responded to the command, it starts sending message data. Each time a message is generated in the Message Processor T5 object, the device automatically sends a message packet to the host with the data. The message packets have the format given in Figure 10-10.

Figure 10-10.Message Packet Format

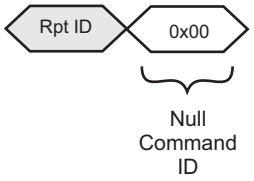


In Figure 10-10:

- **Rpt ID** is the HID-I^{2}C Report ID used for Object Protocol commands and responses (see Table 10-1 on page 36).
- **ID Bytes** identify the packet as an auto-return message packet.
- **Msg ID** is the Report ID returned by the Message Processor T5 object.
- **Message Data** bytes are the bytes of data returned by the Message Processor T5 object. The size of the data depends on the source object for which this is the message data. Refer to the *mXT640U 1.0 Protocol Guide* for more information.

To stop the sending of the messages, the host can send a null command packet. This consists of two bytes: a HID-I^{2}C report ID and a command byte of 0x00 (see Figure 10-11).

Figure 10-11.Null Command Packet Format

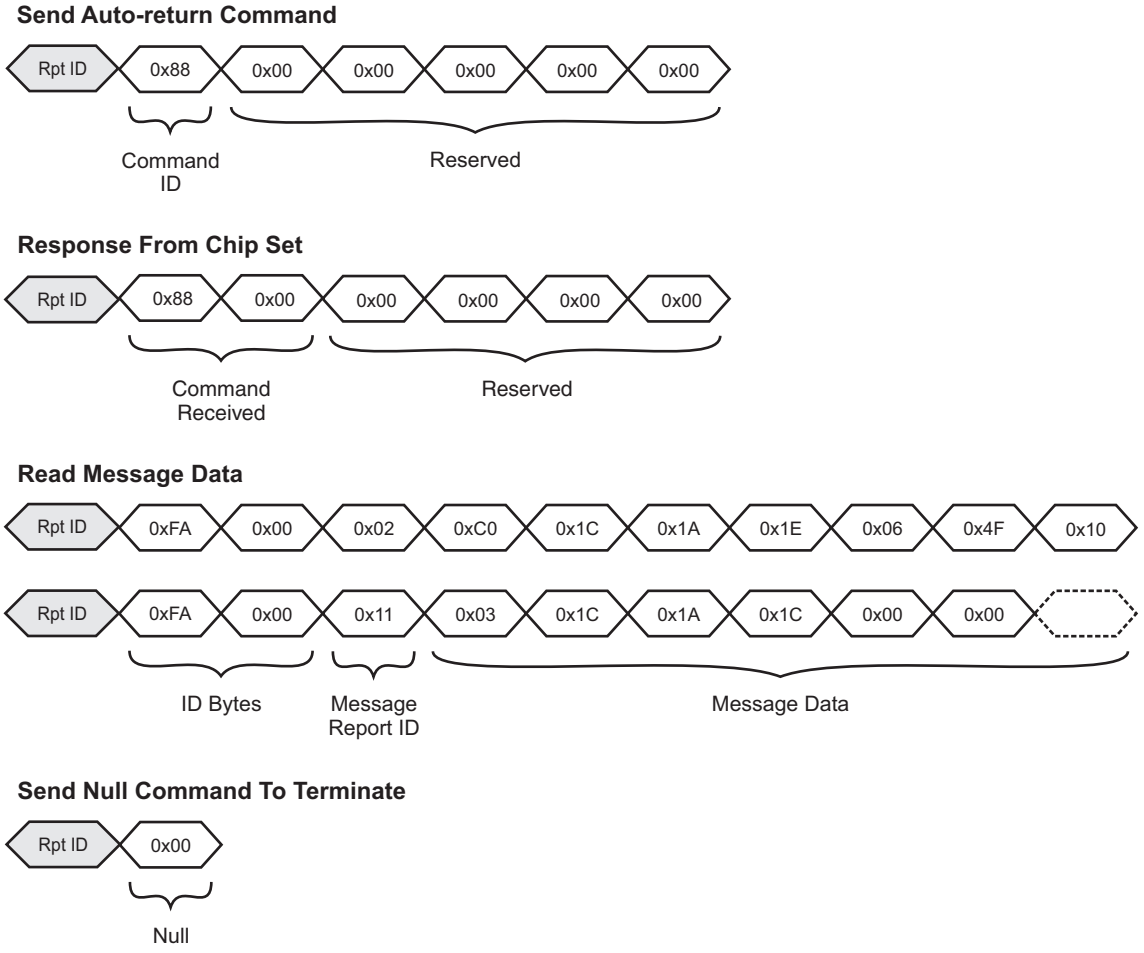


Note that any read or write will also terminate any currently enabled auto-return mode (see Section 10.5.1.2 on page 38).

10.5.2.1 Reading Status Messages

Figure 10-12 shows an example sequence of packets to receive messages from the Message Processor T5 object using the “Send Auto-return” command.

Figure 10-12.Example Auto-return Command Packet



10.6 Digitizer HID-I²C

This is a digitizer class HID.

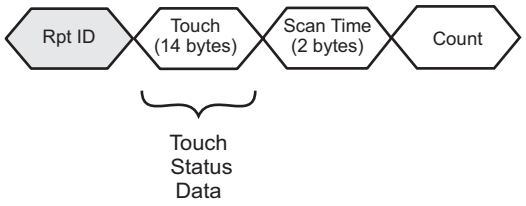
10.6.1 Touch Report

The format of a Touch report is shown in Figure 10.6.2.

Each Touch report starts with a report ID and contains the data for one touch.

The HID-I²C Report ID used is that for Touch reports; see Table 10-1 on page 36 for the value.

10.6.2 Touch Report Packet Format

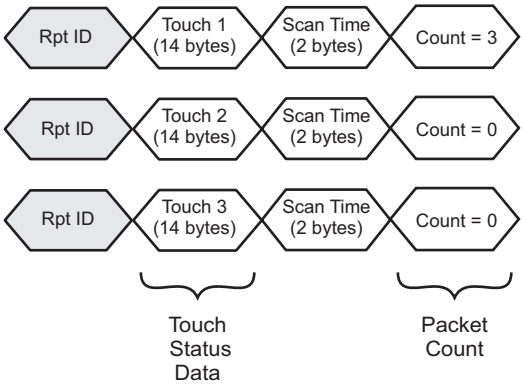


In Figure 10.6.2:

- **Rpt ID** is the HID-I²C Report ID used for Touch reports (see Table 10-1 on page 36).
- **Touch** is the data for the touch.
- **Scan Time** is the Timestamp for the report packet
- **Count** is used to identify the report packets for current active touches that are to be reported as a single package. The Count in the first packet for the first touch is set to the number of active touches to be sent in one package. Subsequent packets for subsequent active touches have a Count of 0.

An example of the Touch report packets for 3 active touches is shown in Figure 10-13.

Figure 10-13.Example Touch Report Packets for 3 Active Touches



Each input report consists of a HID-I²C report ID followed by 17 bytes that describe the status of one active touch. The input report format depends on the geometry calculation control (TCHGEOMEN) of the Digitizer HID Configuration T43 object. Table 10-2 on page 43 and Table 10-3 on page 43 give the detailed format of a touch report packet.

Table 10-2. Touch Report Format when TCHGEOMEN = 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I²C Touch Report ID							
1	Reserved							Status
2	Touch ID							
3 – 4	X Position							
5 – 6	X' Position							
7 – 8	Y Position							
9 – 10	Y' Position							
11	Touch Width							
12	Reserved							
13	Touch Height							
14	Reserved							
15 – 16	Scan Time							
17	Count							

Table 10-3. Touch Report Format when TCHGEOMEN = 0

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	HID-I²C Touch Report ID							
1	Reserved							Status
2	Touch ID							
3 – 4	X Position							
5 – 6	X' Position							
7 – 8	Y Position							
9 – 10	Y' Position							
11	Reserved							
12	Reserved							
13	Reserved							
14	Reserved							
15 – 16	Scan Time							
17	Count							

- **Byte 0:**
  The HID-I²C Report ID (see Table 10-1 on page 36 for Touch reports).
- **Byte 1:**
  Status: Status of the touch detection. This bit is set to 1 if touch is detected, and set to 0, if no touches are detected.
- **Byte 2:**
  Touch ID: Identifies the touch for which this is a status report (starting from 0).

- **Bytes 3 to 10:**

  X and Y positions: These are scaled to 12-bit resolution. This means that the upper four bits of the MSByte will always be zero.
- **Byte 11:**

  Touch Width: Reports the width of the detected touch when TCHGEOMEN is set to 1.
  Reserved when TCHGEOMEN is set to 0
- **Byte 13:**

  Touch Height: Reports the height of the detected touch when TCHGEOMEN is set to 1.
  Reserved when TCHGEOMEN is set to 0
- **Byte 15 to 16:**

  Scan Time: Timestamp associated with the current report packet with a 10 kHz resolution.
- **Byte 17:**

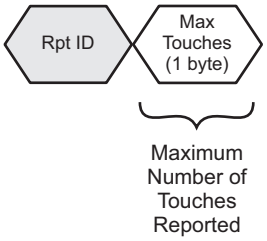
  Count: For the first touch, the number of active touches to be sent in one package. Subsequent packets for subsequent active touches have a Count of 0.

10.6.3 Maximum Touches (Surface Contacts) Report

The format of the Maximum Touches report packet is shown in Figure 10-14.

The HID-I^{2}C Report ID used is that for Maximum Touches reports; see Table 10-1 on page 36 for the value.

Figure 10-14.Example Maximum Touches Report



In Figure 10-14:

- **Rpt ID** is the HID-I^{2}C Report ID used for Maximum Touches reports (see Table 10-1 on page 36).
- **Max Touches** is the maximum number of touches that can be reported by the device.

Read this report to receive the maximum number of touches that can currently be reported.

Write this report to set the maximum number of touches to be reported. Note that the number of touches cannot be set to more than the maximum number of touches defined by Multiple Touch Touchscreen T100 NUMTCH.

10.6.4 Touch Hardware Quality Assurance (THQA) Report

The THQA data is reported to Microsoft Windows using the THQA Report ID (see Table 10-1 on page 36 for the value). The content of this data is defined by Microsoft.

10.7 $\overline{\text{CHG}}$ Line

The $\overline{\text{CHG}}$ line is used to implement the HID-I^{2}C interrupt line. It provides a level triggered interrupt to the host to indicate when there is one or more reports to be read. The $\overline{\text{CHG}}$ line will be pulled low when a report is ready and will remain low as long as there are further reports to be read. Once the last report is read the $\overline{\text{CHG}}$ line will go high.

Note: In order to comply with the HID-I^{2}C specification, Communications Configuration T18 MODE should be set to 0. Refer to the *mXT640U 1.0 Protocol Guide* for more information.

10.8 SDA, SCL

Identical to standard I^{2}C operation. See Section 9.7 on page 34.

10.9 Clock Stretching

Identical to standard I^{2}C operation. See Section 9.8 on page 35.

10.10 Power Control

The mXT640U supports the use of the HID-I^{2}C *SET POWER* commands to put the device into a low power state

10.11 Microsoft Windows Compliance

The mXT640U has algorithms within the Multiple Touch Touchscreen T100 object specifically to ensure Microsoft Windows 8.x compliance.

The device also supports Microsoft Touch Hardware Quality Assurance (THQA) in the Serial Data Command T68 object. Refer to the Microsoft whitepaper *How to Design and Test Multitouch Hardware Solutions for Windows 8*.

These, and other device features, may need specific tuning.

11. PCB Design Considerations

11.1 Introduction

The following sections give the design considerations that should be adhered to when designing a PCB layout for use with the mXT640U. Of these, power supply and ground tracking considerations are the most critical.

By observing the following design rules, and with careful preparation for the PCB layout exercise, designers will be assured of a far better chance of success and a correctly functioning product.

11.2 Printed Circuit Board

Atmel recommends the use of a four-layer printed circuit board for mXT640U applications. This, together with careful layout, will ensure that the board meets relevant EMC requirements for both noise radiation and susceptibility, as laid down by the various national and international standards agencies.

11.2.1 PCB Cleanliness

Modern no-clean-flux is generally compatible with capacitive sensing circuits.

CAUTION: If a PCB is reworked to correct soldering faults relating to any device, or to any associated traces or components, be sure that you fully understand the nature of the flux used during the rework process. Leakage currents from hygroscopic ionic residues can stop capacitive sensors from functioning. If you have any doubts, a thorough cleaning after rework may be the only safe option.

11.3 Power Supply

11.3.1 Supply Quality

While the device has good Power Supply Rejection Ratio properties, poorly regulated and/or noisy power supplies can significantly reduce performance.

Always operate the device with a well-regulated and clean AVdd supply. It supplies the sensitive analog stages in the device.

11.3.2 Supply Rails and Ground Tracking

Power supply and clock distribution are the most critical parts of any board layout. Because of this, it is advisable that these be completed before any other tracking is undertaken. After these, supply decoupling, and analog and high speed digital signals should be addressed. Track widths for all signals, especially power rails should be kept as wide as possible in order to reduce inductance.

The Power and Ground planes themselves can form a useful capacitor. Flood filling for either or both of these supply rails, therefore, should be used where possible. It is important to ensure that there are no floating copper areas remaining on the board: all such areas should be connected to the ground plane. The flood filling should be done on the outside layers of the board.

11.3.3 Power Supply Decoupling

Decoupling capacitor should be fitted as specified in Section 3.2 on page 14.

In addition, at least one “bulk” decoupling capacitor should be placed on each power rail, close to where the supply enters the board.

Surface mounting capacitors are preferred over wire-leaded types due to their lower ESR and ESL. It is often possible to fit these decoupling capacitors underneath and on the opposite side of the PCB to the digital ICs. This will provide the shortest tracking, and most effective decoupling possible.

Refer to the application note *Selecting Decoupling Capacitors for Atmel PLDs* (doc0484.pdf; available on the Atmel website) for further general information on decoupling capacitors.

11.4 Voltage Regulators

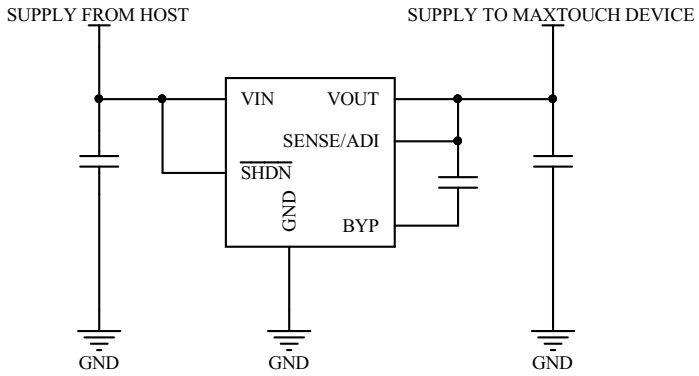
Each supply rail requires a Low Drop-Out (LDO) voltage regulator, although an LDO can be shared where supply rails share the same voltage level.

An LDO regulator should be chosen that provides adequate output capability, low noise, good load regulation and step response. Suitable fixed output LDO devices are shown in Table 11-1.

With a single regulator, PCB layout is more critical than with multiple LDO regulators, and special care with the PCB layout should be taken. See Section 11.4 for information concerning PCB design with a single LDO.

Figure 11-1 shows an example circuit for an LDO.

Figure 11-1. Example LDO Circuit



11.4.1 Multiple Voltage Regulator Supply

The AVdd supply stability is critical for the device because this supply interacts directly with the analog front end. If noise problems exist when using a single LDO regulator, Atmel recommends that AVdd is supplied by a regulator that is separate from the digital supply. This reduces the amount of noise injected into the sensitive, low signal level parts of the design.

11.4.2 Suggested Voltage Regulators

The voltage regulators listed in Table 11-1 have been tested and found to work well with maXTouch devices

Table 11-1. Suitable LDO Regulators

Manufacturer	Device	Current Rating (mA)
Analog Devices	ADP122/ADP123	300
Diodes Inc.	AP2125	300
Diodes Inc.	AP7335	300
Linear Technology	LT1763CS8-3.3	500
NXP	LD6836	300
Texas Instruments	LP3981	300

1. Some manufacturers claim that minimal or no capacitance is required for correct regulator operation. However, in all cases, a minimum of a 1.0 µF ceramic, low ESR capacitor at the input and output of these devices should be used. The manufacturer's datasheets should always be referred to when selecting capacitors for these devices and the typical recommended values, types and dielectrics adhered to.

11.4.3 LDO Selection Criteria

The LDO devices in Table 11-1 on page 47 have been proved to provide satisfactory performance in Atmel maxTouch controllers, however, if it is desired to use an alternative LDO, certain performance criteria should be verified before using the device. These are:

- Stable with low value multi-layer ceramic capacitors on input and output – actual values will be device dependent, but it is good design practice to use values greater than the minimum specified in the LDO regulator data sheet
- Low output noise – less than 100 µV RMS over the range 10 Hz to 1 MHz
- Good load transient response – this should be less than 35 mV peak when a load step change of 100 mA is applied at the device output terminal
- Input supply voltage to match the system requirements
- Low quiescent current to improve battery life
- Thermal and current limit overload protection
- Ideally, select an LDO with common footprint, to allow interchanging between regulators

11.4.4 Single Supply Operation

When designing a PCB for an application using a single LDO, extra care should be taken to ensure short, low inductance traces between the supply and the touch controller supply input pins. Ideally, tracking for the individual supplies should be arranged in a star configuration, with the LDO at the junction of the star. This will ensure that supply current variations or noise in one supply rail will have minimum effect on the other supplies. In applications where a ground plane is not practical, this same star layout should also apply to the power supply ground returns.

Only regulators with a 300 mA or greater rating can be used in a single-supply design.

11.5 I^2C Line Pull-up Resistors

The values for pull-up resistors on SDA and SCL need to be chosen to ensure rise times are within I^2C specification – if the rise time is too long the overall clock rate will be reduced.

If using a VddIO at the low end of the allowable range it is likely that the pull-up resistor values will need to be reduced from those shown on the schematic.

11.6 Analog I/O

In general, tracking for the analog I/O signals from the device should be kept as short as possible. These normally go to a connector which interfaces directly to the touchscreen.

Ensure that adequate ground-planes are used. An analog ground plane should be used in addition to a digital one. Care should be taken to ensure that both ground planes are kept separate and are connected together only at the point of entry for the power to the PCB. This is usually at the input connector.

11.7 Component Placement and Tracking

It is important to orient all devices so that the tracking for important signals (such as power and clocks) are kept as short as possible.

11.7.1 Digital Signals

In general, when tracking digital signals, it is advisable to avoid sharp directional changes on sensitive signal tracks (such as analog I/O) and any clock or crystal tracking.

A good ground return path for all signals should be provided, where possible, to ensure that there are no discontinuities.

11.8 EMC and Other Observations

The following recommendations are not mandatory, but may help in situations where particularly difficult EMC or other problems are present:

- Try to keep as many signals as possible on the inside layers of the board. If suitable ground flood fills are used on the top and bottom layers, these will provide a good level of screening for noisy signals, both into and out of the PCB.
- Ensure that the on-board regulators have sufficient tracking around and underneath the devices to act as a heatsink. This heatsink will normally be connected to the 0 V or ground supply pin. Increasing the width of the copper tracking to any of the device pins will aid in removing heat. There should be no solder mask over the copper track underneath the body of the regulators.
- Ensure that the decoupling capacitors, especially high capacity ceramic type, have the requisite low ESR, ESL and good stability/temperature properties. Refer to the regulator manufacturer's datasheet for more information.

12. Getting Started with mXT640U

12.1 Establishing Contact

12.1.1 Communication with the Host

The host can use the following interface to communicate with the device:

- I^2C interface (see Section 9. on page 30)
- HID-I^2C interface (see Section 10. on page 36)

12.1.2 Power-up Sequence

On power-up, the $\overline{CHG}$ line goes low to indicate that there is new data to be read from the Message Processor T5 object. If the $\overline{CHG}$ line does not go low, there is a problem with the device.

The host should attempt to read any available messages to establish that the device is present and running following power-up or a reset. Examples of messages include reset or calibration messages. The host should also check that there is no configuration error reported.

12.2 Using the Object Protocol

The device has an object-based protocol that is used to communicate with the device. Typical communication includes configuring the device, sending commands to the device, and receiving messages from the device. Refer to the *mXT640U 1.0 Protocol Guide* for more information.

The host must perform the following initialization so that it can communicate with the device:

1. Read the start positions of all the objects in the device from the Object Table and build up a list of these addresses.
2. Use the Object Table to calculate the report IDs so that messages from the device can be correctly interpreted.

12.2.1 Classes of Objects

The mXT640U contains the following classes of objects:

- **Debug objects** – provide a raw data output method for development and testing.
- **General objects** – required for global configuration, transmitting messages and receiving commands.
- **Touch objects** – operate on measured signals from the touch sensor and report touch data.
- **Signal processing objects** – process data from other objects (typically signal filtering operations).
- **Support objects** – provide additional functionality on the device.

12.2.2 Object Instances

Table 12-1. Objects on the mXT640U

Object	Description	Number of Instances	Usage
Debug Objects			
Diagnostic Debug T37	Allows access to diagnostic debug data to aid development.	1	Debug commands only. No configuration/tuning necessary. Not for use in production.
General Objects			
Message Processor T5	Handles the transmission of messages. This object holds a message in its memory space for the host to read.	1	No configuration necessary.

Table 12-1. Objects on the mXT640U (Continued)

Object	Description	Number of Instances	Usage
Command Processor T6	Performs a command when written to. Commands include reset, calibrate and backup settings.	1	No configuration necessary.
Power Configuration T7	Controls the sleep mode of the device. Power consumption can be lowered by controlling the acquisition frequency and the sleep time between acquisitions.	1	Must be configured before use.
Acquisition Configuration T8	Controls how the device takes each capacitive measurement.	1	Must be configured before use.
Touch Objects			
Key Array T15	Creates a rectangular array of keys. A Key Array T15 object reports simple on/off touch information.	1	Enable and configure as required.
Multiple Touch Touchscreen T100	Creates a Touchscreen that supports the tracking of more than one touch.	1	Enable and configure as required.
Signal Processing Objects			
Grip Suppression T40	Suppresses false detections caused, for example, by the user gripping the edge of the touchscreen.	1	Enable and configure as required.
Touch Suppression T42	Suppresses false detections caused, for example, by the user placing their face too near the touchscreen on a mobile phone.	1	Enable and configure as required.
Passive Stylus T47	Processes passive stylus input.	1	Enable and configure as required.
Shieldless T56	Allows a sensor to use true single-layer co-planar construction.	1	Enable and configure as required.
Lens Bending T65	Compensates for lens deformation (lens bending) by attempting to eliminate the disturbance signal from the reported deltas.	3	Enable and configure as required.
maXCharger T72	Performs various noise reduction techniques during touchscreen signal acquisition.	1	Enable and configure as required.
Glove Detection T78	Allows for the reporting of glove touches.	1	Enable and configure as required.
Retransmission Compensation T80	Limits the negative effects on touch signals caused by poor device coupling to ground.	1	Enable and configure as required.
Unlock Gesture T81	Sends a message when a gesture satisfies the configuration settings for use in wake up or unlock situations.	2	Enable and configure as required.
Touch Sequence Processor T93	Captures a sequence of touch and release locations to allow double taps to be detected.	1	Enable and configure as required.
Self Capacitance maXCharger T108	Suppresses the effects of external noise within the context of self capacitance touch measurements.	1	Enable and configure as required.
Self Capacitance Grip Suppression T112	Allows touches to be reported from the self capacitance measurements while the device is being gripped.	1	Enable and configure as required.

Table 12-1. Objects on the mXT640U (Continued)

Object	Description	Number of Instances	Usage
Symbol Gesture Processor T115	Detects arbitrary shaped gestures as a series of ordinal strokes. These are typically symbols drawn by the user for interpretation by the host as wake-up gestures or other application triggers.	1	Enable and configure as required.
Sensor Correction T121	Allows adjustments to be made to mutual measurements from an associated touchscreen sensor.	1	Enable and configure as required.
Support Objects			
Communications Configuration T18	Configures additional communications behavior for the device.	1	Check and configure as necessary.
GPIO/PWM Configuration T19	Creates a bank of digital I/O pins. These pins can then be controlled from the host by writing to the object's configuration memory space.	1	Enable and configure as required.
Self Test T25	Configures and performs self-test routines to find faults on a touch sensor.	1	Configure as required for pin test commands.
User Data T38	Provides a data storage area for user data.	1	Configure as required.
Digitizer HID Configuration T43	Configures the Digitizer HID interface and the Descriptors associated with it.	1	Enable and configure as required.
Message Count T44	Provides a count of pending messages.	1	Read-only object.
CTE Configuration T46	Controls the capacitive touch engine for the device.	1	Must be configured.
Timer T61	Provides control of a timer.	6	Enable and configure as required.
Serial Data Command T68	Provides an interface for the host driver to deliver various data sets to the device.	1	Enable and configure as required.
Dynamic Configuration Controller T70	Allows rules to be defined that respond to system events.	20	Enable and configure as required.
Dynamic Configuration Container T71	Allows the storage of user configuration on the device that can be selected in run-time based on rules defined in the Dynamic Configuration Controller T70 object.	1	Configure if Dynamic Configuration Controller T70 is in use.
CTE Scan Configuration T77	Configures enhanced X line scanning features.	1	Enable and configure as required.
Touch Event Trigger T79	Configures touch triggers for use with the event handler.	3	Enable and configure as required.
Auxiliary Touch Configuration T104	Allows the setting of self capacitance gain and thresholds for a particular measurement to generate auxiliary touch data for use by other objects.	1	Enable and configure if using self capacitance measurements
Self Capacitance Global Configuration T109	Provides configuration for a self capacitance measurements employed on the device.	1	Check and configure as required (if using self capacitance measurements).
Self Capacitance Tuning Parameters T110	Provides configuration space for a generic set of settings for self capacitance measurements.	12	Use under the guidance of Atmel field engineers only.

Table 12-1. Objects on the mXT640U (Continued)

Object	Description	Number of Instances	Usage
Self Capacitance Configuration T111	Provides configuration for self capacitance measurements employed on the device.	2	Check and configure as required (if using self capacitance measurements).
Self Capacitance Measurement Configuration T113	Configures self capacitance measurements to generate data for use by other objects.	1	Enable and configure as required.
Symbol Gesture Configuration T116	Stores configuration data that defines the symbols to be detected by the Symbol Gesture Processor T115 object.	1	Configure if Symbol Gesture Processor T115 is in use.

12.2.3 Configuring and Tuning the Device

The objects are designed such that a default value of zero in their fields is a “safe” value that typically disables functionality. The objects must be configured before use and the settings written to the non-volatile memory using the Command Processor T6 object.

Perform the following actions for each object:

1. Enable the object, if the object requires it.
2. Configure the fields in the object, as required.
3. Enable reporting, if the object supports messages, to receive messages from the object.

Refer also to the *mXT640U 1.0 Protocol Guide* for detailed information on the configuration parameters for the objects.

12.3 Writing to the Device

There are a number of mechanisms for writing to the device:

- Using an I^2C write operation (see Section 9.2 on page 30).
- Using the Generic HID-I^2C write operation (see Section 10.5.1.2 on page 38).

To communicate with the device, you write to the appropriate object:

- To send a command to the device, you write the appropriate command to the Command Processor T6 object (refer to the *mXT640U 1.0 Protocol Guide*).
- To configure the device, you write to an object. For example, to configure the device power consumption you write to the global Power Configuration T7 object, and to set up a touchscreen you write to a Multiple Touch Touchscreen T100 object. Some objects are optional and need to be enabled before use. Refer to the *mXT640U 1.0 Protocol Guide* for more information on the objects.

Important! For I^2C communications, when the host issues any command within an object [1] that results in a flash write to the device Non-Volatile Memory (NVM), that object should have its CTRL RPTEN bit set to 1, if it has one. This ensures that a message from the object writing to the NVM is generated at the completion of the process and an assertion of the $\overline{\text{CHG}}$ line is executed. The host must also ensure that the assertion of the $\overline{\text{CHG}}$ line refers to the expected object report ID before asserting the $\overline{\text{RESET}}$ line to perform a reset. Failure to follow this guidance may result in a corruption of device configuration area and the generation of a CFGERR.

1. For example, from the following objects: Command Processor T6, Serial Data Command T68, Self Capacitance Global Configuration T109.

12.4 Reading from the Device

Status information is stored in the Message Processor T5 object. This object can be read to receive any status information from the device. The following mechanisms provide an interrupt-style interface for reading messages in the Message Processor T5 object:

- The $\overline{\text{CHG}}$ line is asserted whenever a new message is available in the Message Processor T5 object (see Section 9.6 on page 33). See Section 9.4 on page 31 for information on the format of the I^2C read operation.
- When using the HID-I^2C interface, the interface provides an interrupt-driven interface that sends the messages automatically (see Section 10.5.1.3 on page 38)

Note that the host should always wait to be notified of messages. The host should not poll the device for messages.

13. Debugging and Tuning

13.1 SPI Debug Interface

This interface is used for tuning and debugging when running the system and allows the development engineer's PC running Atmel maXTouch Studio to read the real-time raw data using the low-level debug port. This can be accessed via the SPI interface.

The SPI Debug Interface consists of the $\overline{\text{DBG\_SS}}$, DBG_CLK, and DBG_DAT lines. It is recommended that these pins are routed to test points on all designs such that they can be connected to external hardware during system development. These lines should not be connected to power or GND.

The touch controller will take care of the pin configuration. When these lines are in use for debugging, any alternative function for the pins cannot be used.

The SPI Debug Interface is enabled by the Command Processor T6 object and by default will be off.

Refer to the following for more information:

- *mXT640U 1.0 Protocol Guide* for information on the Command Processor T6 object
- QTAN0050, *Using the maXTouch Debug Port*, for information on using the debug port

13.2 Secondary Debug Interface

This interface is used for low-level debugging when developing the system.

The interface consists of the DBG2_CLK, DBG2_FRAME and DBG2_DATA0 to DBG2_DATA5 lines.

These pins should be routed to test points on designs where a new sensor or display technology is being used, or if the design will use an active stylus. These lines should not be connected to power or GND.

The touch controller will take care of the pin configuration. When these lines are in use, any alternative function for the pins cannot be used.

The secondary interface is enabled by a special firmware build and by default will be off.

13.3 Object-based Protocol

The device provides a mechanism for obtaining debug data for development and testing purposes by reading data from the Diagnostic Debug T37 object. Refer to the *mXT640U 1.0 Protocol Guide* for more information on this object.

Note that the Diagnostic Debug T37 is of most use for simple tuning purposes. When debugging a design, it is preferable to use the Primary SPI debug interface, as this will have a much higher bandwidth and can provide real time data.

13.4 Self Test

There is also a Self Test T25 object that runs self-test routines in the device to find hardware faults on the sense lines and the electrodes. This object also performs an initial pin fault test on power-up to ensure that there is no X-to-Y short before the high-voltage supply is enabled inside the chip. A high-voltage short on the sense lines would break the device.

Refer to the *mXT640U 1.0 Protocol Guide* for more information on the Self Test T25 object.

14. Specifications

14.1 Absolute Maximum Specifications

Vdd	3.6 V
VddIO	3.6 V
AVdd	3.6 V
XVdd	10.0 V
Voltage forced onto any pin	–0.3 V to (Vdd, VddIO or AVdd) + 0.3 V
Configuration parameters maximum writes (flash memory write cycles)	10,000

CAUTION: Stresses beyond those listed under *Absolute Maximum Specifications* may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum specification conditions for extended periods may affect device reliability.

14.2 Recommended Operating Conditions

Operating temp	–40°C to +85°C
Storage temp	–60°C to +150°C
Vdd	3.3 V
VddIO	1.8 V to 3.3 V
AVdd	3.3 V
XVdd with internal voltage tripler	Vdd to 3 × Vdd
Cx transverse load capacitance per node	0.6 pF to 3 pF
Temperature slew rate	10°C/min

14.2.1 DC Characteristics

Note: Operation at low supply voltages (AVdd and Vdd) may result in decreased Signal-to-Noise Ratio (SNR). This can cause a reduction in system performance, such as for passive stylus.

14.2.1.1 Analog Voltage Supply

Parameter	Min	Typ	Max	Units	Notes
AVdd					
Operating limits	2.7	3.3	3.6	V	
Supply Rise Rates	–	–	0.25	V/µs	

14.2.1.2 Digital Voltage Supply

Parameter	Min	Typ	Max	Units	Notes
Vdd					
Operating limits	2.7	3.3	3.6	V	
Supply Rise Rates	–	–	0.25	V/µs	
VddIO					
Operating limits	1.71	–	3.47	V	I^2C
Supply Rise Rates	–	–	0.25	V/µs	

14.3 Test Configuration

The values listed below were used in the reference unit to validate the interfaces and derive the characterization data provided in the following sections.

See *mXT640U 1.0 Protocol Guide* for information about the individual objects and their fields.

The values for the user application will depend on the circumstances of that particular project and will vary from those listed here. Further tuning will be required to achieve an optimal performance.

Table 14-1. Test Configuration

Object/Parameter	Description/Setting (Numbers in Decimal)
Acquisition Configuration T8	
CHRGTIME	44
MEASALLOW	11
MEASIDLEDEF	8
MEASACTVDEF	2
GPIO/PWM Configuration T19	Object Enabled
CTE Configuration T46	
IDLESYNCSPERX	16
ACTVSYNCSPERX	16
Shieldless T56	Object Enabled
INTTIME	27
Lens Bending T65 Instance 0	Object Enabled
maXCharger T72	Object Enabled
CTE Scan Configuration T77	Object Enabled
Glove Detection T78	Object Enabled
Retransmission Compensation T80	Object Enabled
Multiple Touch Touchscreen T100	Object Enabled
XSIZE	32
YSIZE	20
Auxiliary Touch Configuration T104	Object Enabled
Self Capacitance maXCharger T108	Object Enabled
Self Capacitance Configuration T111 Instance 0	
INTTIME	50
IDLESYNCSPERL	24
ACTVSYNCSPERL	24

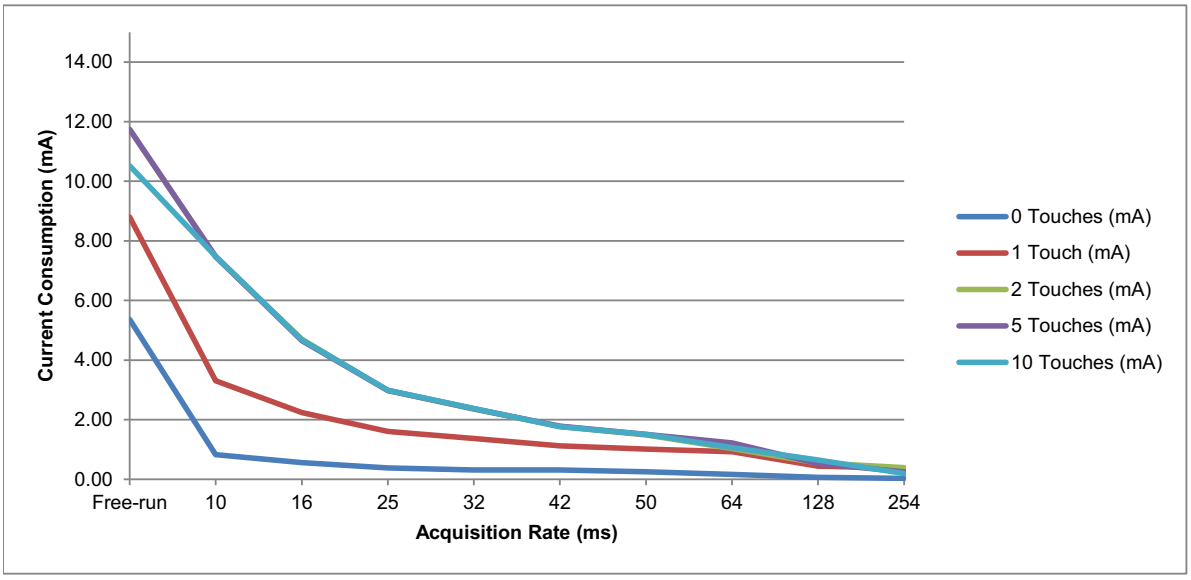
Table 14-1. Test Configuration (Continued)

Object/Parameter	Description/Setting (Numbers in Decimal)
Self Capacitance Configuration T111 Instance 1	
INTTIME	50
IDLESYNCSPERL	32
ACTVSYNCSPERL	32
Self Capacitance Measurement Configuration T113	Object Enabled

14.4 Supply Current

14.4.1 Analog Supply

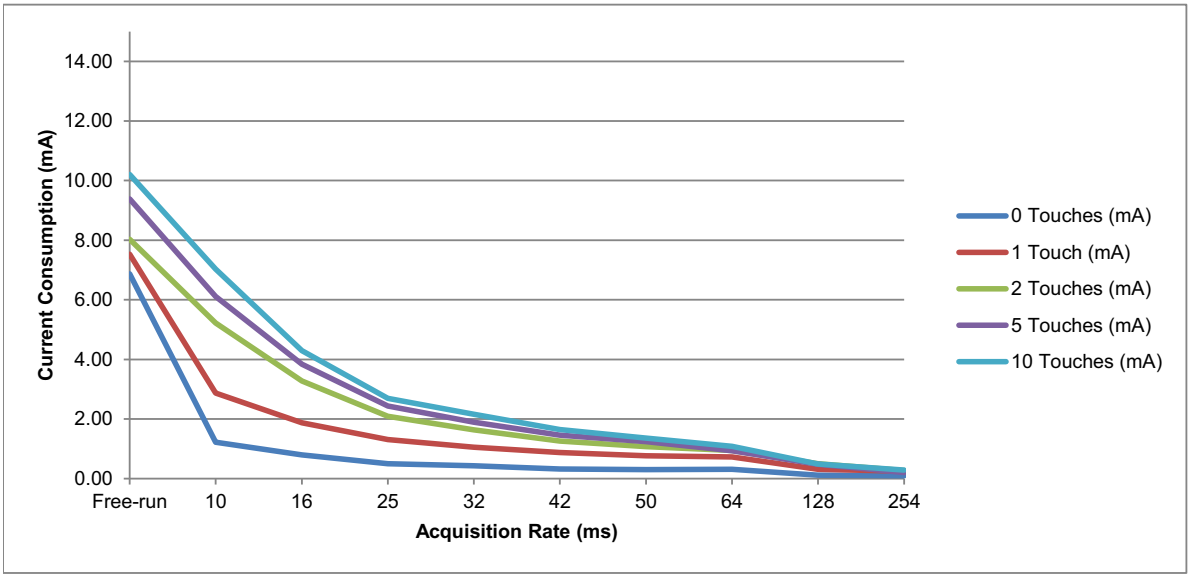
Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	5.36	8.80	11.74	11.74	10.52
10	0.82	3.30	7.46	7.47	7.49
16	0.56	2.24	4.70	4.65	4.68
25	0.38	1.61	2.98	2.98	2.99
32	0.31	1.37	2.36	2.37	2.38
42	0.31	1.13	1.78	1.78	1.77
50	0.26	1.02	1.49	1.50	1.51
64	0.17	0.93	1.03	1.23	1.06
128	0.07	0.45	0.55	0.53	0.65
254	0.03	0.38	0.39	0.25	0.19



14.4.2 Digital Supply

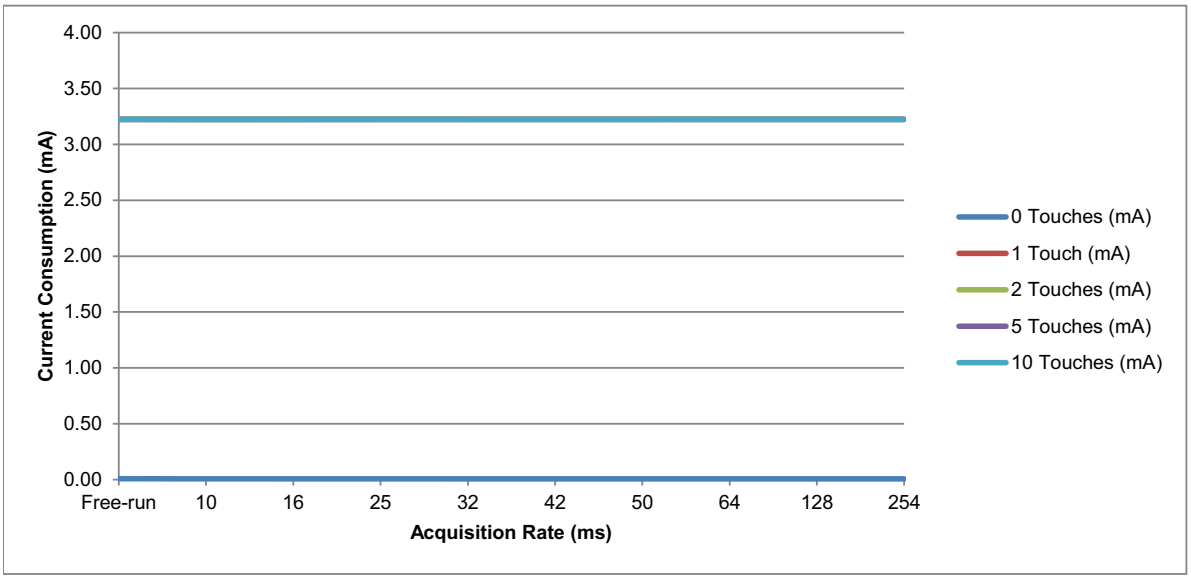
14.4.2.1 Vdd

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	6.87	7.53	8.03	9.39	10.21
10	1.22	2.88	5.23	6.11	7.03
16	0.80	1.87	3.28	3.84	4.29
25	0.50	1.31	2.09	2.44	2.69
32	0.43	1.06	1.64	1.89	2.16
42	0.33	0.88	1.26	1.46	1.65
50	0.30	0.77	1.07	1.24	1.36
64	0.31	0.73	0.95	0.94	1.09
128	0.11	0.31	0.50	0.48	0.49
254	0.10	0.25	0.27	0.21	0.29



14.4.2.2 VddIO

Acquisition Rate (ms)	0 Touches (mA)	1 Touch (mA)	2 Touches (mA)	5 Touches (mA)	10 Touches (mA)
Free-run	0.01	3.22	3.22	3.22	3.22
10	0.01	3.22	3.22	3.22	3.22
16	0.01	3.22	3.22	3.22	3.22
25	0.01	3.22	3.22	3.22	3.22
32	0.01	3.22	3.22	3.22	3.22
42	0.01	3.22	3.22	3.22	3.22
50	0.01	3.22	3.22	3.22	3.22
64	0.01	3.22	3.22	3.22	3.22
128	0.01	3.22	3.22	3.22	3.22
254	0.01	3.22	3.22	3.22	3.22



14.5 Deep Sleep Current

T_A = 25°C

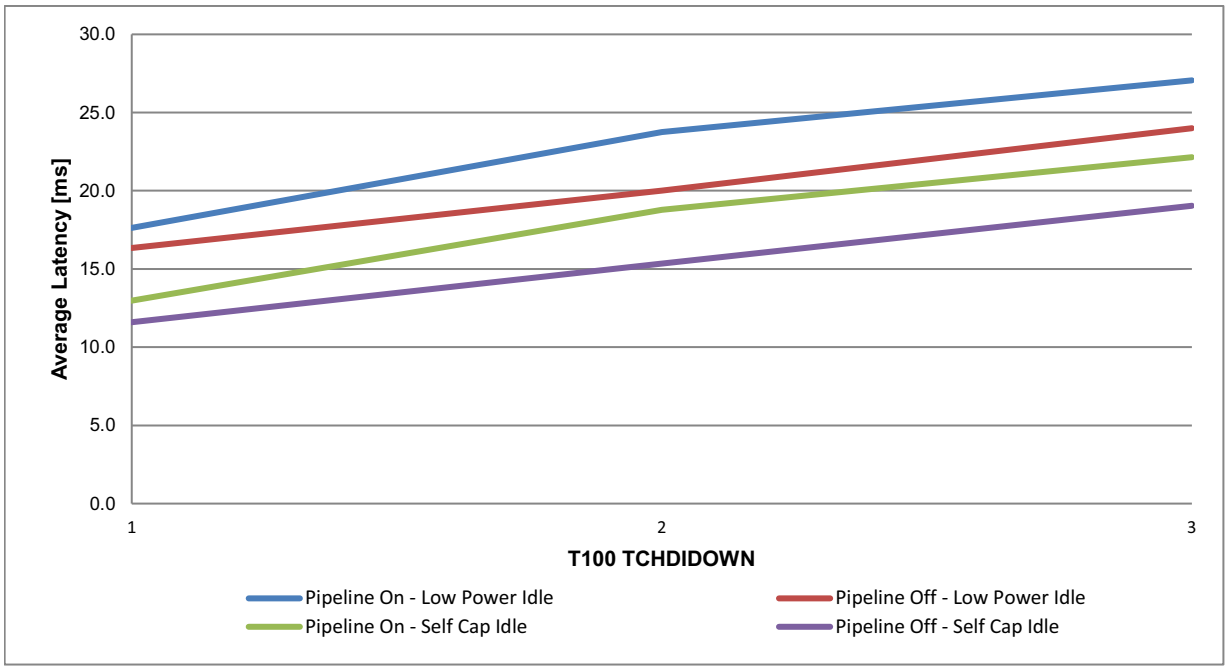
Parameter	Min	Typ	Max	Units	Notes
Deep Sleep Current	–	47	–	µA	Vdd = 3.3 V, AVdd = 3.3 V
Deep Sleep Power	–	150	–	µW	Vdd = 3.3 V, AVdd = 3.3 V

14.6 Power Supply Ripple and Noise

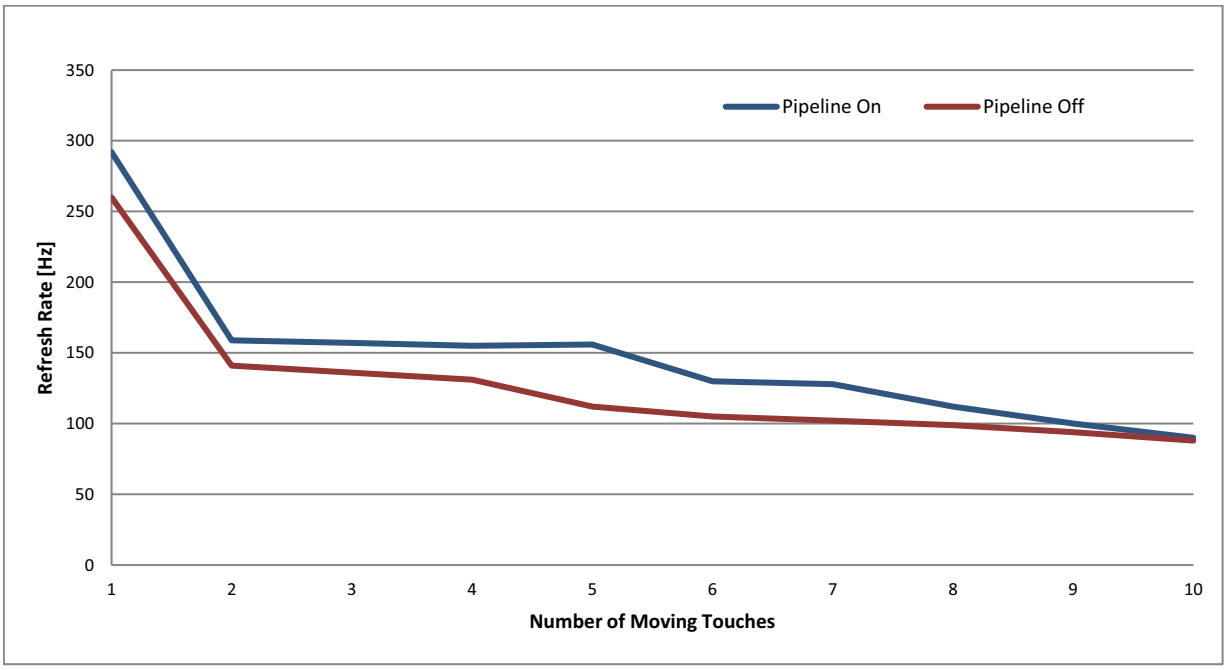
Parameter	Min	Typ	Max	Units	Notes
Vdd	–	–	±50	mV	Across frequency range 1 Hz to 1 MHz
AVdd	–	–	±25	mV	Across frequency range 1 Hz to 1 MHz
AVdd (with noise suppression enabled)	–	–	± 40	mV	Across frequency range 1 Hz to 1 MHz, with Noise Suppression enabled

14.7 Timing Specifications

14.7.1 Touch Latency



14.7.2 Refresh Rate



14.7.3 Reset Timings

Parameter	Min	Typ	Max	Units	Notes
Power on to $\overline{\text{CHG}}$ line low	–	38	–	ms	Vdd supply for POR VddIO supply for external reset
Hardware reset to $\overline{\text{CHG}}$ line low	–	38	–	ms	
Software reset to $\overline{\text{CHG}}$ line low	–	57	–	ms	

Notes:
1. Any $\overline{\text{CHG}}$ line activity before the power-on or reset period has expired should be ignored by the host. Operation of this signal cannot be guaranteed before the power-on/reset periods have expired.
2. The mXT640U meets Microsoft Windows 8.x requirements.

14.8 Input/Output Characteristics

Parameter	Description	Min	Typ	Max	Units	Notes
Input ($\overline{\text{RESET}}$, GPIO, SDA, SCL)						
Vil	Low input logic level	–0.3	–	0.3 × VddIO	V	VddIO = 1.8 V to Vdd
Vih	High input logic level	0.7 × VddIO	–	VddIO	V	VddIO = 1.8 V to Vdd
Iil	Input leakage current	–	–	1	µA	Pull-up resistors disabled
$\overline{\text{RESET}}$ pin	Internal pull-up resistor	9		16	kΩ	
Output ($\overline{\text{CHG}}$, GPIO, SDA, SCL)						
Vol	Low output voltage	0	–	0.2 × VddIO	V	VDDIO = 1.8 V to VDD. Iol = –2 mA
Voh	High output voltage	0.8 × VddIO	–	VddIO	V	VDDIO = 1.8 V to VDD. Ioh = 2 mA

14.9 I²C Specifications

Parameter	Value
Addresses	0x4A or 0x4B
Maximum bus speed (SCL)	3.4 MHz
I²C specification	Version 6.0
Required pull-up resistance for standard mode (100 kHz)	1 kΩ to 10 kΩ (1)
Required pull-up resistance for fast mode (400 kHz)	1 kΩ to 3 kΩ (1)
Required pull-up resistance for fast+ mode (1 MHz)	0.7 kΩ (max) (1)
Required pull-up resistance for high-speed mode (3.4 MHz)	0.5 kΩ to 0.75 kΩ (1)

Notes:
1. The values of pull-up resistors should be chosen to ensure SCL and SDA rise and fall times meet the I²C specification. The value required will depend on the amount of stray capacitance on the line.
2. In systems with heavily laden I²C lines, bus speed may limited by loading and minimum pull-up resistor values.

More detailed information on I²C operation is available from www.nxp.com/documents/user_manual/UM10204.pdf.

14.10 HID-I²C Specification

Parameter	Value
Vendor ID	0x03EB (Atmel)
Product ID	0x2163 (mXT640U)
HID-I²C specification	1.0

14.11 Touch Accuracy and Repeatability

Parameter	Min	Typ	Max	Units	Notes
Linearity (touch only; 5.4 mm electrode pitch)	–	±1	–	mm	8 mm or greater finger
Linearity (touch only; 4.2 mm electrode pitch)	–	±0.5	–	mm	4 mm or greater finger
Accuracy	–	±1	–	mm	
Accuracy at edge	–	±2	–	mm	
Repeatability	–	±0.25	–	%	X axis with 12-bit resolution

14.12 Thermal Packaging

14.12.1 Thermal Data

Parameter	Typ	Unit	Condition	Package

14.12.2 Junction Temperature

The average chip junction temperature, T_J in °C can be obtained from the following:

$$T_J = T_A + (P_D \times \theta_{JA})$$

If a cooling device is required, use this equation:

$$T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$$

where:

- θ_{JA}= package thermal resistance, Junction to ambient (°C/W).
- θ_{JC} = package thermal resistance, Junction to case thermal resistance (°C/W).
- $\theta_{HEATSINK}$ = cooling device thermal resistance (°C/W), provided in the cooling device datasheet.
- P_D = device power consumption (W).
- T_A is the ambient temperature (°C).

14.13 ESD Information

Parameter	Value	Reference standard
Human Body Model (HBM)	±2000 V	JEDEC JS–001
Charge Device Model (CDM)	±250 V	

14.14 Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to Peak)	3°C/s max
Preheat Temperature 175°C ±25°C	150 – 200°C
Time Maintained Above 217°C	60 – 150 s
Time within 5°C of Actual Peak Temperature	30 s
Peak Temperature Range	260°C
Ramp down Rate	6°C/s max
Time 25°C to Peak Temperature	8 minutes max

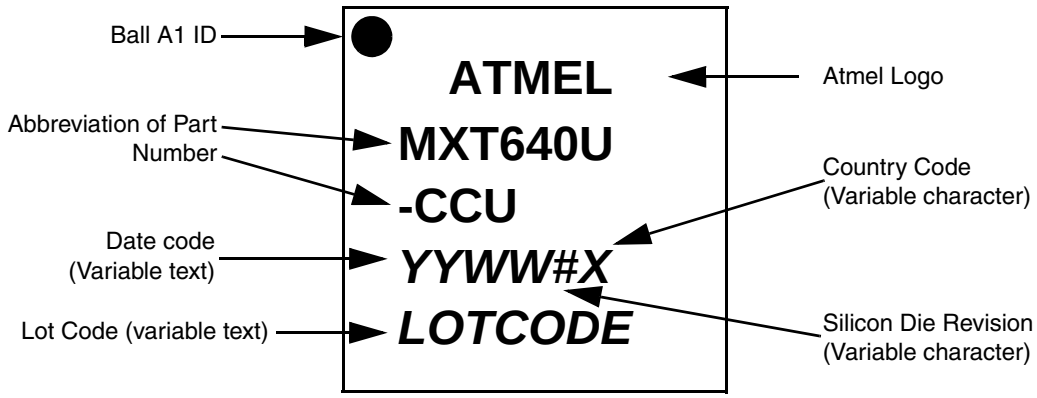
14.15 Moisture Sensitivity Level (MSL)

MSL Rating	Package Type(s)	Peak Body Temperature	Specifications
MSL3	BGA	260°C	IPC/JEDEC J-STD-020

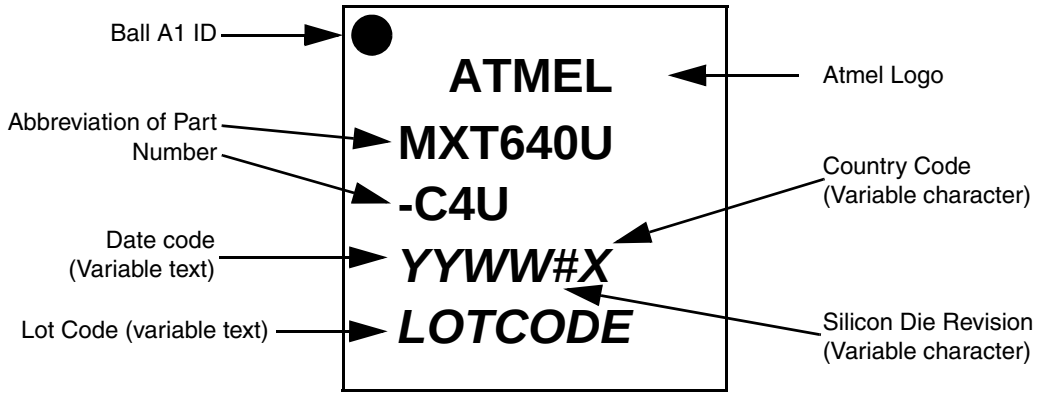
15. Package Information

15.1 Part Markings

15.1.1 ATMXT640U–CCU – UFBGA 88 Balls



15.1.2 ATMXT640U–C4U – X1FBGA 88 Balls



15.2 Orderable Part Numbers

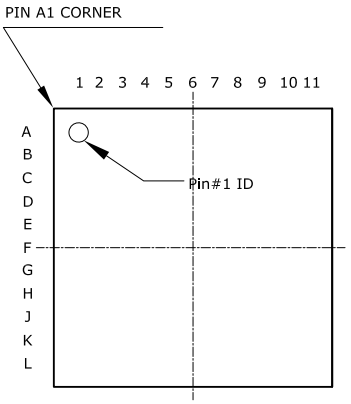
Orderable Part Number	SL Code[1]	Firmware Revision	Description
ATMXT640U-CCU (Supplied in trays)	Q1014	1.0	88-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT640U-CCUR (Supplied in tapes and reels)	Q1014	1.0	88-ball UFBGA 6 × 6 × 0.6 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT640U-C4U (Supplied in trays)	Q1014	1.0	88-ball X1FBGA 6 × 6 × 0.45 mm, 0.5 mm ball pitch, RoHS compliant
ATMXT640U-C4UR (Supplied in tapes and reels)	Q1014	1.0	88-ball X1FBGA 6 × 6 × 0.45 mm, 0.5 mm ball pitch, RoHS compliant

Notes: 1. All purchase orders must include the Orderable Part Number and the SL Code. (Note: SL Code was formerly known as QS Number.)

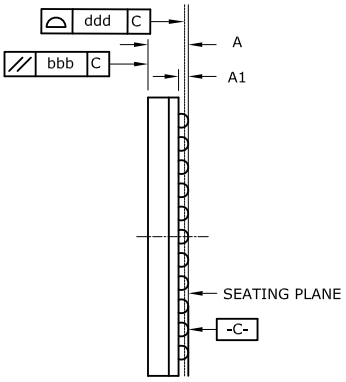
15.3 Mechanical Drawing

15.3.1 UFBGA 88 Balls

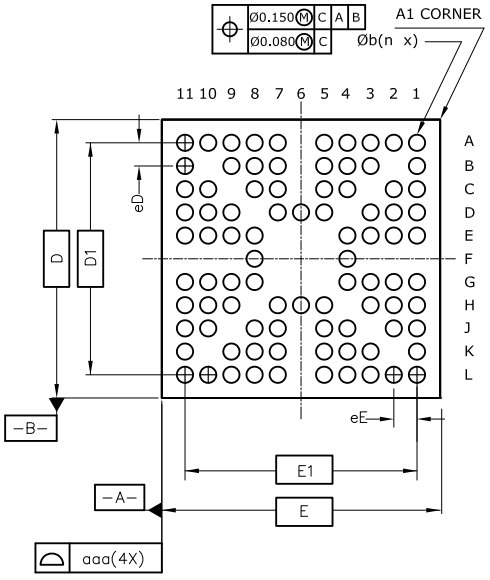
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	0.600	
A1	0.110	-----	0.210	
E/D	6.00 / 6.00			
E1/D1	5.00 / 5.00			
b	0.220	-----	0.280	
eD/eE	Ball pitch : 0.500			
aaa	Pack edge tolerance : 0.100			
bbb	Mold flatness : 0.100			
ddd	Copla : 0.100			
ball diam	0.250			
n	88			

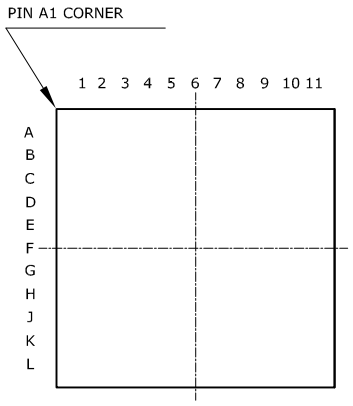
Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-280, Variation UECBA-1 for proper dimensions, tolerances, datums, etc.
2. Array as seen from the bottom of the package.
3. Dimension A includes stand-off height A1, package body thickness, and lid height, but does not include attached features.
4. Dimension b is measured at the maximum ball diameter, parallel to primary datum C.

01/13/2015

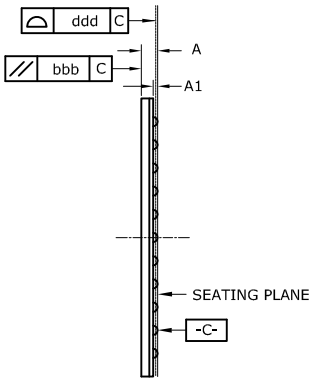
Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	TITLE	GPC	DRAWING NO.	REV.
	76, 88-ball (11x11 Custom, 4P Array), 0.5mm pitch, 6x6x0.6mm Ultra Thin Fine-Pitch Ball Grid Array Package (UFBGA)	CJM	76	A

15.3.2 X1FBGA 88 Balls

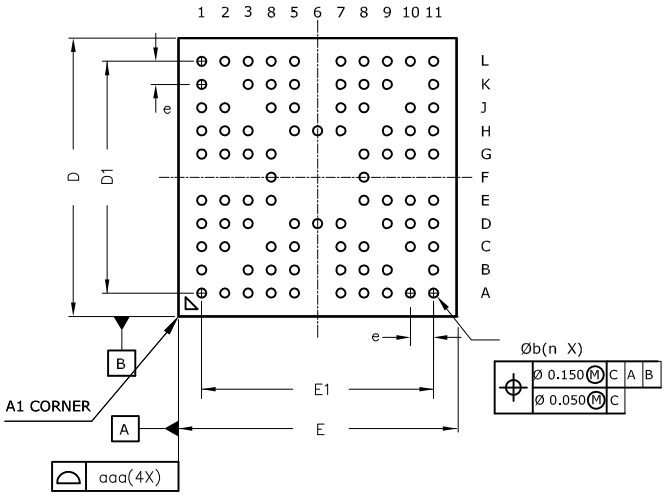
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	0.450	3
A1	0.070	0.100	0.130	
E/D	6.00 / 6.00			
E1/D1	5.00 / 5.00			
b	0.250	0.300	0.350	4
e	Ball pitch : 0.500			
aaa	Pack.edge tolerance : 0.100			
bbb	Mold flatness : 0.100			
ddd	Copla : 0.100			
ball diam	0.200			
n	88			

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-280, Variation UECBA-1 for proper dimensions, tolerances, datums, etc. (Excepted Over-all Height and Nominal Ball Size).
2. Array as seen from the bottom of the package.
3. Dimension A includes stand-off height A1, package body thickness, and lid height, but does not include attached features.
4. Dimension b is measured at the maximum ball diameter, parallel to primary datum C.

02/15/2016

Atmel	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: **packagedrawings@atmel.com**	88-ball (11x11 Custom), 0.5mm pitch, 6x6x0.45mm Extra-Thin Fine-Pitch Ball Grid Array Package (X1FBGA)	CJL	XB	A

Associated Documents

The following documents are available by contacting Atmel Touch Technology Division:

Note: Most of the documents listed below are available under NDA only. In addition, some documents may have further restrictions placed upon them.

- **Product Documentation:**
  - Application Note: MXTAN0213 – *Interfacing with Atmel maXTouch Controllers*
  - *mXT640U 1.0 Protocol Guide* (distributed with Atmel approval only)
- **Touchscreen design and PCB/FPCB layout guidelines:**
  - Application Note: QTAN0054 – *Getting Started with maXTouch Touchscreen Designs*
  - Application Note: MXTAN0208 – *Design Guide for PCB Layouts for Atmel Touch Controllers*
  - Application Note: QTAN0080 – *Touchscreens Sensor Design Guide*
  - Application Note: doc0484 – *Selecting Decoupling Capacitors for Atmel PLDs* (doc0484.pdf; available on the Atmel website)
- **Configuring the device:**
  - Application Note: QTAN0059 – *Using the maXTouch Self Test Feature*
  - Application Note: MXT0202 – *Using the Unlock Gesture T81 Object*
- **Miscellaneous:**
  - Application Note: QTAN0050 – *Using the maXTouch Debug Port*
  - Application Note: QTAN0058 – *Rejecting Unintentional Touches with the maXTouch Touchscreen Controllers*
  - Application Note: QTAN0061 – *maXTouch Sensitivity Effects for Mobile Devices*
  - Application Note: QTAN0051 – *Bootloading Procedure for Atmel Touch Sensors Based on the Object Protocol*
- **Tools:**
  - *maXTouch Studio User Guide* (distributed as on-line help with maXTouch Studio)

Revision History

Revision Number	History
Revision ASX – February 2015	Initial edition for firmware revision 1.0 – Advance
Revision BSX – July 2015	Updated – Summary
Revision CX – November 2015	Updated for general distribution
Revision DX – November 2015	Updated for general distribution – Preliminary • Schematic updated to show 10 kΩ pull-up resistor for $\overline{\text{CHG}}$ • Specifications updated. Xdd now correctly reads Vdd not Avdd. Internal pull-up resistor for $\overline{\text{RESET}}$ specified.
Revision EX – December 2015	Updated – Release
Revision FX – April 2016	• X1FBGA (6 × 6 × 0.45 mm) package added • X2FBGA (6 × 6 × 0.4 mm) package removed • *Schematic Notes*: subsections reorganized. Notes on I^2C interface, multiple function pins, GPIO pins, debug lines added • *Circuit Components*: section removed. Content moved to *Schematic Notes* and *PCB Design Consideration* sections • *PCB Design Consideration*: sections added from deleted *Circuit Components)* section • Other minor edits to aid readability

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel München GmbH
Business Campus
Parkring 4
D-85748 Garching bei München
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370